# INPUT 64

DAS ELEKTRONISCHE MAGAZIN 12/85

Infos · News · Programme · Unterhaltung · Tips




INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61






12/85

## *Liebe 64er-Besitzer(in),*

*mit dieser Ausgabe des elektronischen Magazins erhalten Sie INPUT 64 das letzte Mal im gewohnten Gewand, in der Präsentation und mit den Möglichkeiten, die manche von Ihnen seit mittlerweile zwölf Ausgaben kennen. Richtig, wir könnten einjähriges Jubiläum feiern. Aber geschenkt, genauso geschenkt, wie Ihnen zum hunderttausendsten Mal ein "Frohes Fest" oder einen "Guten Rutsch" zuzurufen. Statt guter Wünsche für's neue Jahr gibt es INPUT 64 ab 1986 mit neuem Layout der Bildschirmseiten - Sie erhalten schon beim Durchblättern des Inhalts-Menues ausführlichere Programminformationen. Und statt eines Weihnachtsgeschenks haben wir einen oft geäußerten Leserwunsch verwirklicht: Ab Ausgabe 1/86 können Sie Anwenderprogramme, Tools und Spiele auch im SuperTape-Format auf Kassette überspielen. (Natürlich wahlweise weiterhin im Commodore-Format.)*

*Und um Ihnen unangenehme Überraschungen zu ersparen, kündigen wir branchenunüblich, aber nach Verlagsgepflogenheit die Preiserhöhung ab der nächsten Ausgabe jetzt schon an. Der Einzelpreis für die Kassette beträgt künftig 14,80 DM, der Preis für die Diskettenversion wird nicht erhöht ! Damit bleibt diese Preissteigerung deutlich unter der Kostensteigerung. Abonnenten sparen noch mehr als bislang: Auch der Preis für das INPUT 64-Abo wird nicht erhöht! Ab Januar also: 14,80 DM für die Kassette, 19,80 DM für die Diskette, der Preis fürs Abo gleichbleibend bei 140,– DM (Kassette) beziehungsweise 198,– DM (Diskette). Das Abo ist nach wie vor jederzeit kündbar.*

*Ihre INPUT 64-Redaktion*

### Auf einen Blick: INPUT 64-Betriebssystembefehle

| | |
|---|---|
| Inhaltsverzeichnis aufrufen | CTRL und i |
| Hilfsseite aufrufen | CTRL und h |
| Programm abspeichern | CTRL und s |
| Bildschirmausdruck | CTRL und b |
| Titelbild abkürzen | CTRL und q |

Laden von Kassette mit LOAD oder SHIFT+RUN/STOP
Laden von Diskette mit LOAD "INPUT*",8,1
Ausführliche Bedienungshinweise finden Sie auf Seite 29.

# Leser fragen

## BASIC auf und ab

*In der Praxis funktioniert der Hinweis im letzten Heft ganz gut: die Programme aus Ihrem Magazin, die nach dem Laden bei LIST scheinbar nur aus einer Zeile bestehen, nach dem Start aber das vollständige Listing zeigen, durch den Befehl POKE 44,8 nach einer eventuellen Änderung wieder abgespeichert werden können. (Es geht um die Anfrage unter der Überschrift "INKEY abgehängt?", d.Red.) Mich würde aber auch interessieren, was da genau geschieht... (tel. Anfrage)*

Der Speicherbereich für BASIC-Programme beginnt normalerweise bei der Adresse 2049 ($0801). Davor liegt der Bildschirm, diverse Zeiger für das Betriebssystem und die sogenannte "Zero-Page", das sind die Adressen, die durch ein Byte (Werte von 0 bis 255) darstellbar sind. Aus zwei Speicherzellen dieser Zero-Page entnimmt der BASIC-Interpreter den Beginn des BASIC-Bereichs, den Adressen 43 und 44 ($2B/$2C). Der Inhalt der Adresse 43 stellt das Low-Byte, der der Adresse 44 das High-Byte des BASIC-Anfangs dar. Nach dem Einschalten des Rechners können Sie hier durch PEEK die Werte 1 beziehungsweise 8 auslesen. Da man diese beiden Zeiger auch durch POKE beschreiben kann, läßt sich der BASIC-Anfang verschieben. Sinnvoll ist dies, um Platz zu schaffen für Sprite-Muster oder Maschinensprache-Routinen. Diese Methode wird auch bei den oben angesprochenen INPUT-Programmen angewandt. Der SYS-Befehl in der ersten Zeile springt zu einem Maschinenprogramm, daß

- den Zeiger 45 verändert und somit den BASIC-Anfang um (meist) drei Pages (=3*256 Byte) verschiebt und - den Befehl RUN ausgibt.

Dadurch wird das BASIC-Programm, daß 'höher' im Speicher liegt, gestartet. Setzen Sie die Adresse 44 wieder auf den Wert 8, gilt wieder der alte BASIC-Anfang, und Sie sehen nur die Zeile mit dem SYS-Aufruf. Falls Sie mit diesen POKEs selbst experimentieren, beachten Sie noch folgendes: die Adresse direkt vor dem BASIC-Anfang muß (!) eine 0 enthalten. Ansonsten reagiert der Rechner auf die Eingabe von Befehlen nur mit SYNTAX-ERROR. Anschließend müssen durch NEW die anderen BASIC-Pointer (auf Programm-Ende und so weiter) neu gesetzt werden. Vergißt man dies, stürzt das System bei der Eingabe einer Zeile heillos ab. Wenn also der BASIC-Bereich erst ab Adresse 2305 ($0901) beginnen soll, geben Sie ein POKE 44,9 (setzt den Bereich eine Page höher) und POKE 2304,0. Dann können Sie im so definierten Speicherbereich ein BASIC-Programm eingeben oder nachladen. Die gewonnenen 256 Byte können beispielsweise zur Ablage von vier Sprite-Definitionen benutzt werden. (d. Red.)

## RELOCATOR oben und unten

*Welche Einsprung-Adresse stimmt für die zweite Version des Relocators aus Ausgabe 10/85: $6000, wie auf dem Bildschirm angegeben, oder $9000, wie im Beiheft steht?*

*(mehrere tel. Anfragen)*

Halten Sie sich an die Angaben auf dem Bildschirm. Die Version des Relocators im BASIC-Bereich beginnt bei Adresse $6000. Aufgerufen wird das Programm dementsprechend mit SYS 6*4096. (d. Red.)

## Rhythmusmaschine

# FUNKY DRUMMER

Machen Sie aus Ihrem C-64 einen digitalen Drum-Computer! Mit diesem über 16 Kilobyte langen Maschinen-Programm wird Ihr C-64 zur digitalen Rhythmus-Maschine. Sie können sich auf dem Schlagzeug begleiten lassen, während Sie Ihr Instrument spielen. Mit etwas Kreativität erstellen Sie sich einfach Ihren eigenen Rhythmus erstellen.

### Der Autor über sich selbst:

"Mein Name ist Peter Fröhlich. Ich wurde am 13.3.1969 in Bad Wimpfen geboren.

Vor mehr als drei Jahren fing ich an, mich für die Telespiele zu interessieren. Mein Interesse galt der Programmierung dieser Art von Spielen. Doch ich mußte mich noch etwas gedulden, denn meinen C 64 bekam ich erst vor ungefähr zweieinhalb Jahren. Dann konnte es aber richtig losgehen. Nach den ersten Programmen in BASIC, merkte ich bald, daß man Spiele auf diese Art nicht effektiv genug programmieren konnte. So befaßte ich mich mit der Maschinen-Sprache. Ein großer Teil meiner Fähigkeiten liegt in der Programmierung von Computer-Grafiken und Musik. So entstand das Programm Funky Drummer.

Übrigens mein Berufswunsch geht auch in die Richtung der Computer-Grafik."

### Wie funktioniert's?

Das Prinzip der digitalen Wiedergabe ist leicht erklärt. Die originalen Instrumentenklänge werden über einen Analog-Digital-Wandler in binäre Werte zerlegt und im Computer abgelegt. Die Wiedergabe der Klänge erfolgt nun gerade umgekehrt, das heißt, die Werte werden vom Programm aus dem Speicher gelesen und über das Lautstärkeregister im SID (Sound-Interface-Device) wieder in natürliche Klänge verwandelt. Zur Programmtechnik siehe Listing.

### Bedienung

Nach dem Starten des Programms sieht man das Titelbild von Funky Drummer. Durch Drücken irgendeiner Taste gelangt man in das Menü. Im Bildabschnitt 2 (Bild 1) werden mit den Cursor-Tasten "links und rechts" die einzelnen Modi angewählt (blauer Pfeil). Mit der RETURN-Taste wird der jeweilige Modus aktiviert (Pfeil wird weiß) und mit der S-Taste verlässt man ihn wieder. Sie können wählen zwischen sechs verschiedenen Modi.

### DRUM-Modus

Hier können die Instrumente mit den Tasten 1 bis 8 direkt gespielt werden. Die entsprechende Taste für jedes Instrument ist in Tabelle 1 aufgeführt. Mit den Tasten F1, F3, F5 und F7 kann die Tonhoehe eingestellt werden, in der das jeweilige Schlaginstrument erklingen soll. Um einen Trommelwirbel zu erzeugen, können während des Spielens die SHIFT- oder die Commodore-Taste gedrückt werden (repeat Funktion).

### EDIT-Modus

In diesem Modus werden die einzelnen Pattern (Rhythmus-Muster) erstellt. Der Cursor (blauer Rahmen) wird mittels der Cursor-Tasten über das Pattbord im Bildabschnitt 1 (Bild 1) bewegt. Am linken Rand zeigt ein Pfeil die aktuelle Pattline (Musterzeile) an. Die insgesamt acht Pattboards mit jeweils acht Pattlines werden mit den Tasten A bis H angewählt und auf der rechten Seite durch einen zweiten Pfeil angezeigt. An der Cursorposition kann nun mit den Tasten 1 bis 8 ein Instrumenten-Symbol gesetzt oder mit der Pfeil-links-Taste wieder gelöscht werden. Mittels der Taste "P" wird die aktuelle Pattline gespielt.

### ARRANGE-Modus

Jetzt wird die Reihenfolge festgelegt, in der die einzelnen Pattlines gespielt werden sollen. Der Cursor (blauer Pfeil) wird wie gewohnt mit den Cursor-Tasten nach links und rechts gesteuert. An der Cursor-Position kann nun mit den Tasten 1 bis 8 und A bis H die gewünschte Pattline eingegeben werden. Die SPACE-Taste löscht eine ungewollte Eingabe wieder. Mit der Sternchen-Taste setzt man ein Wiederholungszeichen, so daß die Wiedergabe von vorn beginnt. Merke: Der Computer spielt die Pattern beziehungsweise Pattline von links nach rechts.

## PLAY-Modus

Die im ARRANGE-Modus geschaffenen Arrangements werden gespielt. Aktivieren Sie diesen Modus indem Sie zusätzlich eine der im DRUM-Modus beschriebenen Tasten drücken, so wird der VIC (Video-Interface-Controller) bis zum Verlassen des Modus abgeschaltet. Das hat den Vorteil, daß der Klang reiner wird, und Störgeräusche vom Fernseher, die durch den hohen Bildkontrast entstehen können, verringert werden. Diese Möglichkeit ist auch im DRUM-Modus anwendbar.

## STORRAGE-Modus

Die erstellten Pattboards, die Arrange-Zeilen, die Klangfarbe und die Geschwindigkeit, in der der von Ihnen erstellte Rhythmus gespielt werden soll, können hier von Diskette beziehungsweise Kassette geladen und auch abgespeichert werden. Auf die Frage "LOAD OR SAVE", ist mit der Taste "L" oder "S" zu antworten. Danach wird der maximal acht Zeichen lange Name eingegeben. Es sind Buchstaben, Zahlen und Leerzeichen als Eingabe gestattet. Mit der Taste DEL kann eine Fehleingabe korrigiert und mit der RETURN-Taste die Eingabe beendet werden. Achtung!! Beim Speichern wird ein unter dem gleichen Namen abgespeichertes File überschrieben. Nach dem Laden beziehungsweise Speichern, gelangt man sofort in das Menü zurück.

## SPEED-Modus

Das ganz rechte Feld im Bildabschnitt 2 ist zum ändern der Drum-Geschwindigkeit. Sie kann mittels der Tasten "CURSOR rauf" und "CURSOR runter" verändert werden. Dabei können Sie Werte von 0 bis 99 einstellen. Merke: Je kleiner der angezeigte Wert ist, desto größer ist die Geschwindigkeit.

Nach einem RESET kann das Programm durch SYS 18432 von neuem gestartet werden. Die vorherigen Einstellungen bleiben dabei erhalten.

Funky Drummer
DRUM EDIT PLAY ARRANGE STORAGE 00
BY BANANASOFT

Bild 1
Abschnitt 1
" 2
" 3
" 4

Ein Beispiel:

Starten Sie das Programm, und drücken Sie, nachdem das Titelbild erschienen ist, irgendeine Taste. Das Menü erscheint. Mit den Cursor-Tasten rechts und links wird der DRUM-Modus ausgewählt. Drückt man nun auf eine der Tasten von 1 bis 8, ertönt das jeweilige Instrument. Wenn Sie von diesem Modus genug haben, betätigen Sie die S-Taste. Gehen Sie jetzt in den EDIT-Modus. Ein blauer Rahmen stellt den Cursor dar, den Sie wiederum mit den Cursor-Tasten bewegen können. Bringen Sie den blauen Rahmen in die erste Zeile (Pattline 1A) und setzen Sie Ihren Ideen entsprechend mit den Tasten 1 bis 8 mehrere Symbole. Genauso verfahren Sie anschließend mit der Zeile zwei. Durch betätigen der Tasten A bis H können Sie ein anderes Pattboard einstellen. Verlassen Sie jetzt diesen Modus mit der S-Taste. Wählen Sie das Feld "ARRANGE". Im Bildabschnitt 4 bewegen Sie den kleinen blauen Pfeil ganz nach links und geben eine 1 und ein A ein. Gehen Sie mit dem Pfeil eine Stelle weiter nach rechts und geben jetzt eine 2 und ein A ein. Anschließend wieder die S-Taste drücken – wenn Sie jetzt den PLAY-Modus aktivieren, können Sie hören, was Sie eingegeben haben.

**Speicherbelegung**

(Nach Start durch RUN)

| Adressbereich | Belegung |
|---|---|
| \$0800 – \$09FF | BASIC-Programm |
| \$0B00 – \$2AFF | abgespeicherte SOUNDS |
| \$3000 – \$3480 | Zeichensatz und Sprites |
| \$4000 – \$43FF | Video-RAM für Menü |
| \$4400 – \$47FF | Farb-RAM für Menü |
| \$4800 – \$5A89 | Maschinen-Programm |
| \$5C00 – \$5FFF | Video-RAM für Titelbild |
| \$6000 – \$7FFF | Bitmap für Titelbild |
| \$8000 – \$83FF | Farb-RAM für Titelbild |

**Tabelle 1**

| Instrument | Taste |
|---|---|
| Bass-Drum | 1 |
| Snare-Drum | 2 |
| Becken | 3 |
| HiHat | 4 |
| Woodblock | 5 |
| Tom-Tom (klein) | 6 |
| Tom-Tom (groß) | 7 |
| Cowbell | 8 |
| Pause | - |

```
;***********************************
;***                             ***
;***     Funky Drummer           ***
;***                             ***
;***                             ***
;***                             ***
;***  PROGRAM & DESIGN : PETER   ***
;***                  FROEHLICH  ***
;***           (C)1985 BY        ***
;***    VERLAG HEINZ HEISE GMBH  ***
;***                             ***
;***  A BANANASOFT PRODUCTION '85 ***
;***********************************

*         =     $4800      ;PRG.START
          JMP   TITLE      ;

;**********        LABELS         **********

V         =     $D000      ;VIC
SI        =     $D400      ;SID
VI1       =     $4000      ;VIDEORAM DATA
VI2       =     $0400      ;VIDEORAM
FA1       =     $4400      ;COLORRAM DATA1
FA2       =     $D800      ;COLORRAM
FA3       =     32576      ;COLORRAM DATA2
PATT      =     $04CC      ;START P.BOARD
PUTT      =     $D8CC      ;START P.BOARD
FILEPARA  =     $FFBA      ;FILEPARAMETER
FILENAME  =     $FFBD      ;NAMEPARAMETER
LOAD      =     $FFD5      ;LOADROUTINE
SAVE      =     $FFD8      ;SAVEROUTINE
FORM1     =     2040       ;SPRITEPOINTER1
FORM2     =     2041       ;      "       2
FORM3     =     2042       ;      "       3
MODE      .BYTE 1          ;TOPICAL MODE
KEY       =     203        ;KEY PRESSED
SHIFT     =     653        ;SHIFT-FLAG
X1        =     V+0        ;ARROW X-POS
MSB       =     V+16       ;SPRITE MSB
D.BD      =     $0B00      ;DATA BASSDRUM
D.SD      =     $0F00      ; "    SNAREDRUM
D.CYM     =     $1300      ; "    CYMBAL
D.HH      =     $1700      ; "    HIHAT
D.WOOD    =     $1B00      ; "    WOODBLOCK
D.TOMHI   =     $1F00      ; "    TOMHI
D.TOMLO   =     $2300      ; "    TOMLO
D.COW     =     $2700      ; "    COWBELL
U1        =     $02        ;USER-BYTE
U2        .BYTE 0          ;RAMBOARD POS

D.BOLO    .BYTE <BOARD1,<BOARD2,<BOARD3
          .BYTE <BOARD4,<BOARD5,<BOARD6
          .BYTE <BOARD7,<BOARD8

D.BOHI    .BYTE >BOARD1,>BOARD2,>BOARD3
          .BYTE >BOARD4,>BOARD5,>BOARD6
          .BYTE >BOARD7,>BOARD8

D.ARRLLO  .BYTE $CA,$F2,$1A,$42
          .BYTE $6A,$92,$BA,$E2

D.ARRLHI  .BYTE $04,$04,$05,$05
          .BYTE $05,$05,$05,$05

D.ARRRLO  .BYTE $EE,$16,$3E,$66
          .BYTE $8E,$B6,$DE,$06

D.ARRRHI  .BYTE $04,$05,$05,$05
          .BYTE $05,$05,$05,$06

D.1       .BYTE 56,59,8,11,16,19,24,27

D.2       .BYTE 10,28,20,18,14,21,26,29

D.LOSA    .BYTE 12,15,1,4,32,15,18,32
          .BYTE 19,1,22,5

D.DIDA    .BYTE 4,9,19,11,32,15,18,32
          .BYTE 20,1,16,5

D.NAME    .BYTE 10,28,20,18,14,21,26,29
          .BYTE 33,34,37,42,36,39,38,41
          .BYTE 62,17,13,22,30,31,9,23
          .BYTE 25,12,35,56,59,8,11,16
          .BYTE 19,24,27,32,60

D.POKE    .BYTE 1,2,3,4,5,6,7,8,9,10,11
          .BYTE 12,13,14,15,16,17,18,19
          .BYTE 20,21,22,23,24,25,26,48
          .BYTE 49,50,51,52,53,54,55,56
          .BYTE 57,32

D.OVER    .BYTE 64,58
D.LINE    .BYTE 68,82,85,77,77,69,82,32
D.FILE    .BYTE 32,32,32,32,32,32,32,32

;************* TITLESCREEN *************

TITLE     LDA   #150       ;SET UP
          STA   56576      ;TITLESCREEN

          BNE   INIT1      ;

          LDX   #$00       ;INIT VIC
INIT2     LDA   D.VIC1,X   ;
          STA   V     ,X   ;
          INX              ;
          CPX   #46        ;
          BNE   INIT2      ;

          LDX   #207       ;SPR.FORM DEF
          STX   FORM1      ;
          INX              ;
          STX   FORM2      ;
          INX              ;
          STX   FORM3      ;

          LDA   #1         ;INIT
          STA   MODE       ;LABELVALUE
          STA   U5         ;
          STA   U6         ;
          STA   U12        ;
          STA   U13        ;
          LDA   #0         ;
          STA   U2         ;
          LDA   #<PATT     ;
          STA   U3         ;
          LDA   #>PATT     ;
          STA   U4         ;
          LDA   #<BOARD1   ;
          STA   U8         ;
          LDA   #>BOARD1   ;
          STA   U9         ;
          JSR   SETUP      ;
          JSR   ARRL       ;
          JSR   ARRR       ;
          JSR   DISPLAY    ;
          LDX   #0         ;
INIT3     LDA   TOP,X      ;
          STA   $0749,X    ;
          LDA   BUT,X      ;
          STA   $0771,X    ;
          INX              ;
          CPX   #38        ;
          BNE   INIT3      ;

;**********       MAINPROGRAM      **********

          LDA   #$0E       ;ARROW = C015
          STA   V+39       ;
MAIN      JSR   SETSPR     ;FIX ARROW
```

```
U3        =     $FB        ;VIDEORAM POS
U4        =     $FC        ;    "       ↑
U5        .BYTE 1          ;CURRENT   LINE
U6        .BYTE 1          ;  "  PATTBOARD
U8        =     $FD        ;CURRENT RAM-
U9        =     $FE        ;BOARD ADRESS
U10       =     $3F        ;USER-BYTE
U11       =     $40        ;USER-BYTE
U12       .BYTE 1          ;CURRENT COLUMN
U13       .BYTE 1          ;CURRENT ARLINE
G01       =     $41        ;DOIT BYTE 1
G02       =     $42        ;     "    2
G03       =     $3D        ;     "    3
G04       =     $3E        ;     "    4

D.START                    ;SAVESTARTPOINT

U14       .BYTE 5          ;DRUMSPEED
U7        .BYTE $37        ;SPEED
LINE      =     $06D1      ;DISPLAYLINE
;**********       DATES        **********

BOARD1    * = * + 256      ;PATTBOARD1
BOARD2    * = * + 256      ;PATTBOARD2
BOARD3    * = * + 256      ;PATTBOARD3
BOARD4    * = * + 256      ;PATTBOARD4
BOARD5    * = * + 256      ;PATTBOARD5
BOARD6    * = * + 256      ;PATTBOARD6
BOARD7    * = * + 256      ;PATTBOARD7
BOARD8    * = * + 256      ;PATTBOARD8

TOP       * = * + 38       ;ARRANGE TOP
BUT       * = * + 38       ;ARRANGE BUTTON

D.END                      ;SAVE ENDPOINT

D.VIC1    .BYTE  60,178, 33,234, 54 ;0
          .BYTE  90,$00,$00,$00,$00 ;5
          .BYTE $00,$00,$00,$00,$00 ;10
          .BYTE $00,$00, 27,$00,$00 ;15
          .BYTE $00,$07,200,$00, 28 ;20
          .BYTE $00,$00,$00,$00,$00 ;25
          .BYTE $00,$00,$00,$00,$00 ;30
          .BYTE $00,$00,$00,$00, 14 ;35
          .BYTE  14,  6,$00,$00,$00 ;40
          .BYTE $00,$00             ;45

D.SPR1    .BYTE  60,108,156,228, 44, 76
D.SPR2    .BYTE   0,  0,  0,  0,  1,  1

          LDA  #59          ;
          STA  V+17         ;
          LDA  #120         ;
          STA  V+24         ;
          LDA  #216         ;
          STA  V+22         ;
          LDA  #0           ;
          STA  V+33         ;
          STA  V+32         ;
          STA  V+21         ;

          LDX  #0           ;
TITLE1    LDA  FA3,X        ;
          STA  FA2,X        ;
          LDA  FA3+256,X    ;
          STA  FA2+256,X    ;
          LDA  FA3+512,X    ;
          STA  FA2+512,X    ;
          LDA  FA3+768,X    ;
          STA  FA2+768,X    ;
          LDA  FA3+744,X    ;
          STA  FA2+744,X    ;
          INX               ;
          BNE  TITLE1       ;
TITLE3    LDA  KEY          ;WAIT ON KEY
          CMP  #64          ;
          BEQ  TITLE3       ;
          LDA  #151         ;
          STA  56576        ;

;************ SET UP RAM *************

INIT      LDX  #$00         ;
INIT1     LDA  VI1      ,X  ;LOAD VIDEORAM
          STA  VI2      ,X  ;
          LDA  VI1+256,X    ;
          STA  VI2+256,X    ;
          LDA  VI1+512,X    ;
          STA  VI2+512,X    ;
          LDA  VI1+744,X    ;
          STA  VI2+744,X    ;
          LDA  FA1      ,X  ;LOAD COLORRAM
          STA  FA2      ,X  ;
          LDA  FA1+256,X    ;
          STA  FA2+256,X    ;
          LDA  FA1+512,X    ;
          STA  FA2+512,X    ;
          LDA  FA1+744,X    ;
          STA  FA2+744,X    ;
          DEX               ;

MAIN4     LDA  KEY          ;
          CMP  #$01         ;RETURN?
          BEQ  MENUE        ;
          CMP  #$02         ;CURSOR <=> ?
          BNE  MAIN4        ;
          LDA  SHIFT        ;
          BEQ  MAIN2        ;
          DEC  MODE         ;MOVE ARROW
          LDA  MODE         ;LEFT
          BNE  MAIN3        ;
          LDA  #$06         ;
          STA  MODE         ;
MAIN3     JMP  MAIN         ;
MAIN2     INC  MODE         ;MOVE ARROW
          LDA  MODE         ;RIGHT
          CMP  #$07         ;
          BNE  MAIN         ;
          LDA  #$01         ;
          STA  MODE         ;
          JMP  MAIN         ;

MENUE     LDA  #$01         ;GO TO MODE
          STA  V+39         ;
          LDA  MODE         ;
          CMP  #$01         ;
          BNE  MODE2        ;
          LDA  #64          ;
          STA  U1           ;
          JMP  DRUMMER      ;DRUMMER
MODE2     CMP  #$02         ;
          BNE  MODE3        ;
          JMP  EDIT         ;EDIT
MODE3     CMP  #$4          ;
          BNE  MODE4        ;
          JMP  SORT         ;ARRANGE
MODE4     CMP  #$03         ;
          BNE  MODE5        ;
          JMP  DOIT         ;PLAY
MODE5     CMP  #$05         ;
          BNE  MODE6        ;
          JMP  STORA        ;STORAGE
MODE6     JMP  SPEED        ;SPEED

;**********    SETSPR(ITE)     **********

SETSPR    LDX  MODE         ;PUT MODE ARROW
          DEX               ;IN RIGHT POS
          LDA  D.SPR1,X     ;
          STA  X1           ;
          LDA  D.SPR2,X     ;
```

```
          TAX                  ;
          LDA   MSB            ;
          AND   #$FE           ;
          CPX   #$00           ;
          BEQ   SETSPR1        ;
          ORA   #$01           ;
SETSPR1   STA   MSB            ;
          LDX   #$60           ;
SETSPR2   LDY   #$00           ;
SETSPR3   DEY                  ;
          BNE   SETSPR3        ;
          DEX                  ;
          BNE   SETSPR2        ;
          RTS                  ;

;**********    DRUM MODE    **********

DRUMMER   LDA   SHIFT          ;
          BEQ   DRUM           ;
          LDA   V+17           ;
          AND   #%11101111;
          STA   V+17           ;
DRUM      LDA   SHIFT          ;DRUM
          BNE   KEY9           ;
          LDA   KEY            ;
          CMP   U1             ;
          BEQ   DRUM           ;
KEY9      LDA   KEY            ;
          STA   U1             ;
          CMP   #56            ;
          BNE   KEY2           ;
          JSR   BD             ;
          JMP   DRUM           ;
KEY2      CMP   #59            ;
          BNE   KEY3           ;
          JSR   SD             ;
          JMP   DRUM           ;
KEY3      CMP   # 8            ;
          BNE   KEY4           ;
          JSR   CYM            ;
          JMP   DRUM           ;
KEY4      CMP   #11            ;
          BNE   KEY5           ;
          JSR   HH             ;
          JMP   DRUM           ;
KEY5      CMP   #16            ;
          BNE   KEY6           ;
          JSR   WOOD           ;
          JMP   DRUM           ;

          INX                  ;
          BNE   BD1            ;
BD2       LDA   D.BD+256,X;
          JSR   OUT            ;
          INX                  ;
          BNE   BD2            ;
BD3       LDA   D.BD+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   BD3            ;
BD4       LDA   D.BD+768,X;
          JSR   OUT            ;
          INX                  ;
          BNE   BD4            ;
          RTS                  ;

SD        LDX   #$00           ;SNARE-DRUM
SD1       LDA   D.SD,X         ;
          JSR   OUT            ;
          INX                  ;
          BNE   SD1            ;
SD2       LDA   D.SD+256,X;
          JSR   OUT            ;
          INX                  ;
          BNE   SD2            ;
SD3       LDA   D.SD+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   SD3            ;
SD4       LDA   D.SD+768,X;
          JSR   OUT            ;
          INX                  ;
          BNE   SD4            ;
          RTS                  ;

CYM       LDX   #$00           ;CYMBAL
CYM1      LDA   D.CYM,X        ;
          JSR   OUT            ;
          INX                  ;
          BNE   CYM1           ;
CYM2      LDA   D.CYM+256,X;
          JSR   OUT            ;
          INX                  ;
          BNE   CYM2           ;
CYM3      LDA   D.CYM+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   CYM3           ;
CYM4      LDA   D.CYM+768,X;

          INX                  ;
          BNE   TOMHI2         ;
TOMHI3    LDA   D.TOMHI+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMHI3         ;
TOMHI4    LDA   D.TOMHI+768,X;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMHI4         ;
          RTS                  ;

TOMLO     LDX   #$00           ;TOM LO
TOMLO1    LDA   D.TOMLO,X      ;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMLO1         ;
TOMLO2    LDA   D.TOMLO+256,X;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMLO2         ;
TOMLO3    LDA   D.TOMLO+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMLO3         ;
TOMLO4    LDA   D.TOMLO+768,X;
          JSR   OUT            ;
          INX                  ;
          BNE   TOMLO4         ;
          RTS                  ;

COW       LDX   #$00           ;COWBELL
COW1      LDA   D.COW,X        ;
          JSR   OUT            ;
          INX                  ;
          BNE   COW1           ;
COW2      LDA   D.COW+256,X;
          JSR   OUT            ;
          INX                  ;
          BNE   COW2           ;
COW3      LDA   D.COW+512,X;
          JSR   OUT            ;
          INX                  ;
          BNE   COW3           ;
COW4      LDA   D.COW+768,X;
          JSR   OUT            ;
          INX                  ;
          BNE   COW4           ;
          RTS                  ;
```

```
KEY6     CMP  #19        ;
         BNE  KEY7       ;
         JSR  TOMHI      ;
         JMP  DRUM       ;
KEY7     CMP  #24        ;
         BNE  KEY8       ;
         JSR  TOMLO      ;
         JMP  DRUM       ;
KEY8     CMP  #27        ;
         BNE  ENDDRUM    ;
         JSR  COW        ;
         JMP  DRUM       ;
ENDDRUM  CMP  #63        ;
         BEQ  END        ;
         CMP  #4         ;F1
         BNE  DRUM1      ;
         LDA  #1         ;
         STA  U14        ;
         JMP  DRUM       ;

DRUM1    CMP  #5         ;F3
         BNE  DRUM2      ;
         LDA  #3         ;
         STA  U14        ;
         JMP  DRUM       ;

DRUM2    CMP  #6         ;F5
         BNE  DRUM3      ;
         LDA  #5         ;
         STA  U14        ;
         JMP  DRUM       ;

DRUM3    CMP  #3         ;F7
         BNE  DRUM4      ;
         LDA  #7         ;
         STA  U14        ;
DRUM4    JMP  DRUM       ;

END      LDA  #$0E       ;
         STA  V+39       ;
         LDA  V+17       ;
         ORA  #%00010000;
         STA  V+17       ;
         JMP  MAIN       ;

;**********    PLAY DRUMS      **********

BD       LDX  #$00       ;BASS-DRUM
BD1      LDA  D.BD,X     ;
         JSR  OUT        ;
         JSR  OUT        ;
         INX             ;
         BNE  CYM4       ;
         RTS             ;

HH       LDX  #$00       ;HI-HAT
HH1      LDA  D.HH,X     ;
         JSR  OUT        ;
         INX             ;
         BNE  HH1        ;
HH2      LDA  D.HH+256,X;
         JSR  OUT        ;
         INX             ;
         BNE  HH2        ;
HH3      LDA  D.HH+512,X;
         JSR  OUT        ;
         INX             ;
         BNE  HH3        ;
HH4      LDA  D.HH+768,X;
         JSR  OUT        ;
         INX             ;
         BNE  HH4        ;
         RTS             ;

WOOD     LDX  #$00       ;WOODBLOCK
WOOD1    LDA  D.WOOD,X   ;
         JSR  OUT        ;
         INX             ;
         BNE  WOOD1      ;
WOOD2    LDA  D.WOOD+256,X;
         JSR  OUT        ;
         INX             ;
         BNE  WOOD2      ;
WOOD3    LDA  D.WOOD+512,X;
         JSR  OUT        ;
         INX             ;
         BNE  WOOD3      ;
WOOD4    LDA  D.WOOD+768,X;
         JSR  OUT        ;
         INX             ;
         BNE  WOOD4      ;
         RTS             ;

TOMHI    LDX  #$00       ;TOM HI
TOMHI1   LDA  D.TOMHI,X  ;
         JSR  OUT        ;
         INX             ;
         BNE  TOMHI1     ;
TOMHI2   LDA  D.TOMHI+256,X;
         JSR  OUT        ;

SPACE    LDA  #4         ;EMPTY SPACE
SPACE3   LDX  #0         ;
SPACE2   JMP  SPACE1     ;
SPACE1   JSR  OUT2       ;
         INX             ;
         BNE  SPACE2     ;
         TAY             ;
         DEY             ;
         TYA             ;
         CMP  #0         ;
         BNE  SPACE3     ;
         RTS             ;

OUT      STA  SI+$18     ;SOUND OUTPUT
OUT2     LDY  U14        ;
OUT1     DEY             ;
         BNE  OUT1       ;
         RTS             ;

;**********      EDIT        **********

EDIT     JSR  SETUP      ;
EDIT3    LDA  KEY        ;
         CMP  #07        ;
         BNE  EDIT1      ;
         LDA  SHIFT      ;CURSOR UP DOWN
         BEQ  EDIT2      ;
         JSR  CURUP      ;
         JMP  EDIT1      ;
EDIT2    JSR  CURDOWN    ;
EDIT1    LDA  KEY        ;
         CMP  #02        ;
         BNE  EDIT4      ;
         LDA  SHIFT      ;CURSOR < = >
         BEQ  EDIT5      ;
         JSR  CURLEFT    ;
         JMP  EDIT4      ;
EDIT5    JSR  CURRIGH    ;
EDIT4    LDA  KEY        ;
         CMP  #63        ;STOP ?
         BNE  EDIT6      ;
         JMP  END        ;
EDIT6    LDX  #0         ;
EDIT7    LDA  KEY        ;
         CMP  D.1,X      ;KEY 1-8
         BEQ  PRINT      ;PRESSED ?
         INX             ;
         CPX  #$08       ;
         BNE  EDIT7      ;
```

```
         CMP  #57          ;PAUSE KEY ←
         BNE  EDIT8        ;PRESSED ?
         LDA  #37          ;
         LDY  #0           ;
         STA  (U3),Y       ;
         LDY  U2           ;
         STA  (U8),Y       ;
         LDY  #0           ;
         LDA  U3           ;
         STA  U10          ;
         LDA  U4           ;
         CLC               ;
         ADC  #$D4         ;
         STA  U11          ;
         LDA  #6           ;
         STA  (U10),Y      ;
         JMP  EDIT8        ;
PRINT    TXA               ;FIX SYMBOL
         CLC               ;
         ADC  #38          ;
         LDY  #0           ;
         STA  (U3),Y       ;
         LDY  U2           ;
         STA  (U8),Y       ;
         LDY  #0           ;
         LDA  U3           ;
         STA  U10          ;
         LDA  U4           ;
         CLC               ;
         ADC  #$D4         ;
         STA  U11          ;
         LDA  #14          ;
         STA  (U10),Y      ;
EDIT8    LDX  #0           ;KEY A-H
EDIT9    LDA  KEY          ;PRESSED
         CMP  D.2,X        ;
         BEQ  CHANGE1      ;
         INX               ;
         CPX  #$08         ;
         BNE  EDIT9        ;
         JMP  EDIT10       ;
CHANGE1  JSR  CHANGE       ;CHANGE
EDIT10   JSR  WAIT         ;PATTBOARD
         LDA  KEY          ;
         CMP  #41          ;
         BNE  EDIT11       ;
         JSR  PLAY         ;
         JSR  WAIT         ;
         JSR  WAIT         ;
         JSR  WAIT         ;

CURLEFT1 DEC  U12          ;
         LDA  V+4          ;
         SEC               ;
         SBC  #8           ;
         STA  V+4          ;
         BCS  CURLEFT3     ;
         LDA  V+16         ;
         AND  #%11111011;
         STA  V+16         ;
CURLEFT3 DEC  U2           ;
         DEC  U3           ;
         LDA  U3           ;
         CMP  #$FF         ;
         BNE  CURLEFT4     ;
         DEC  U4           ;
CURLEFT4 RTS               ;
CURRIGH  LDA  U12          ;MOVE CURSOR
         CMP  #32          ;RIGHT
         BNE  CURRIGH1     ;
         RTS               ;
CURRIGH1 INC  U12          ;
         LDA  V+4          ;
         CLC               ;
         ADC  #8           ;
         STA  V+4          ;
         BCC  CURRIGH3     ;
         LDA  V+16         ;
         ORA  #%00000100;
         STA  V+16         ;
CURRIGH3 INC  U2           ;
         INC  U3           ;
         LDA  U3           ;
         CMP  #$00         ;
         BNE  CURRIGH4     ;
         INC  U4           ;
CURRIGH4 RTS               ;

CHANGE   INX               ;PATTBOARD
         STX  U6           ;CHANGE
         DEX               ;
         LDA  D.BOLO,X     ;
         STA  U8           ;
         LDA  D.BOHI,X     ;
         STA  U9           ;
         JSR  SETUP        ;
         JSR  ARRR         ;
         RTS               ;

WAIT     LDX  #$30         ;WAIT

         BNE  SETUP7       ;
         LDA  #$06         ;
         STA  PUTT+256,X;
SETUP7   INX               ;
         BNE  SETUP6       ;

         RTS               ;READY
PLUS8    LDA  U10          ;
         CLC               ;
         ADC  #8           ;
         STA  U10          ;
         BCC  PLUS8.1      ;
         INC  U11          ;
PLUS8.1  RTS               ;

;**********  SET ARROW LEFT  **********
ARRL     LDX  #$00         ;CLEAR ALL
ARRL1    LDA  D.ARRLLO,X;ARROWS ON
         STA  U10          ;LEFT SIDE
         LDA  D.ARRLHI,X;
         STA  U11          ;
         LDY  #00          ;
         LDA  #32          ;
         STA  (U10),Y      ;
         INX               ;
         CPX  #$08         ;
         BNE  ARRL1        ;

         LDX  U5           ;SET CURRENT
         DEX               ;ARROW
         LDA  D.ARRLLO,X;
         STA  U10          ;
         LDA  D.ARRLHI,X;
         STA  U11          ;
         TYA               ;
         STA  (U10),Y      ;
         RTS               ;

;**********  SET ARROW RIGHT **********

ARRR     LDX  #$00         ;CLEAR ALL
ARRR1    LDA  D.ARRRLO,X;ARROWS ON
         STA  U10          ;RIGHT SIDE
         LDA  D.ARRRHI,X;
         STA  U11          ;
         LDY  #00          ;
         LDA  #32          ;
         STA  (U10),Y      ;
```

```
EDIT11   JMP  EDIT3        ;
CURUP    LDA  U5           ;MOVE CURSOR UP
         CMP  #1           ;
         BNE  CURUP1       ;
         RTS               ;
CURUP1   DEC  U5           ;
         JSR  ARRL         ;
         LDA  V+5          ;
         SEC               ;
         SBC  #8           ;
         STA  V+5          ;
         LDA  U2           ;
         SEC               ;
         SBC  #32          ;
         STA  U2           ;
         LDA  U3           ;
         SEC               ;
         SBC  #40          ;
         STA  U3           ;
         BCS  CURUP2       ;
         DEC  U4           ;
CURUP2   RTS               ;

CURDOWN  LDA  U5           ;MOVE CURSOR
         CMP  #8           ;DOWN
         BNE  CURDOWN1     ;
         RTS               ;
CURDOWN1 INC  U5           ;
         JSR  ARRL         ;
         LDA  V+5          ;
         CLC               ;
         ADC  #8           ;
         STA  V+5          ;
         LDA  U2           ;
         CLC               ;
         ADC  #32          ;
         STA  U2           ;
         LDA  U3           ;
         CLC               ;
         ADC  #40          ;
         STA  U3           ;
         BCC  CURDOWN2     ;
         INC  U4           ;
CURDOWN2 RTS               ;

CURLEFT  LDA  U12          ;MOVE CURSOR
         CMP  #1           ;LEFT
         BNE  CURLEFT1     ;
         RTS               ;

WAIT2    LDY  #$00         ;
WAIT1    DEY               ;
         BNE  WAIT1        ;
         DEX               ;
         BNE  WAIT2        ;
         RTS               ;

;********** SETUP DRUM BOARD **********

SETUP    LDX  #<PATT       ;SETUP CURRENT
         STX  U10          ;PATTBOARD
         LDX  #>PATT       ;
         STX  U11          ;
         LDY  #$00         ;
SETUP2   LDX  #$00         ;
SETUP1   LDA  (U8) ,Y      ;
         STA  (U10),Y      ;
         INY               ;
         BEQ  SETUP8       ;
         INX               ;
         CPX  #32          ;
         BNE  SETUP1       ;
         JSR  PLUS8        ;
         JMP  SETUP2       ;

SETUP8   LDA  #$0E         ;PATTBOARD =
         LDX  #$00         ;LIGHTBLUE
SETUP3   STA  PUTT,X       ;
         STA  PUTT+40,X    ;
         STA  PUTT+80,X    ;
         STA  PUTT+120,X;
         STA  PUTT+160,X;
         STA  PUTT+200,X;
         STA  PUTT+240,X;
         STA  PUTT+280,X;
         INX               ;
         CPX  #32          ;
         BNE  SETUP3       ;

         LDX  #$00         ;ALL PAUSE
SETUP4   LDA  PATT,X       ;CHARS = BLUE
         CMP  #$25         ;
         BNE  SETUP5       ;
         LDA  #$06         ;
         STA  PUTT,X       ;
SETUP5   INX               ;
         BNE  SETUP4       ;
SETUP6   LDA  PATT+256,X;
         CMP  #$25         ;

         INX               ;
         CPX  #$08         ;
         BNE  ARRR1        ;
         LDX  U6           ;SET CURRENT
         DEX               ;ARROW
         LDA  D.ARRRLO,X;
         STA  U10          ;
         LDA  D.ARRRHI,X;
         STA  U11          ;
         TYA               ;
         STA  (U10),Y      ;
         RTS               ;

;********** CHANGE SPEED **********

SPEED    LDA  KEY          ;
         CMP  #$7          ;
         BNE  SPEED1       ;
         CMP  #$7          ;
         LDA  SHIFT        ;
         BEQ  SPEED2       ;
         JSR  UP           ;CURSOR UP
         JMP  SPEED1       ;
SPEED2   JSR  DOWN         ;CURSOR DOWN
SPEED1   LDA  KEY          ;
         CMP  #63          ;STOP ?
         BNE  SPEED3       ;
         JMP  END          ;
SPEED3   LDX  #$20         ;WAIT
SPEED5   LDY  #$00         ;
SPEED4   DEY               ;
         BNE  SPEED4       ;
         DEX               ;
         BNE  SPEED5       ;
         JMP  SPEED        ;

UP       LDA  U7           ;REDUCE SPEED

         CMP  #$99         ;
         BEQ  DISPLAY      ;
         AND  #%00001111;
         TAY               ;
         INY               ;
         CPY  #10          ;
         BNE  UP1          ;
         LDA  U7           ;
         AND  #%11110000;
         CLC               ;
         ADC  #16          ;
```

```
        STA  U7          ;
        JMP  DISPLAY     ;
UP1     STY  U1          ;
        LDA  U7          ;
        AND  #%11110000;
        CLC              ;
        ADC  U1          ;
        STA  U7          ;
DISPLAY LDA  U7          ;
        AND  #%00001111;
        CLC              ;
        ADC  #48         ;
        STA  $067E       ;
        LDA  U7          ;
        AND  #%11110000;
        LSR              ;
        LSR              ;
        LSR              ;
        LSR              ;
        CLC              ;
        ADC  #48         ;
        STA  $067D       ;
        RTS              ;

DOWN    LDA  U7          ;RAISE SPEED
        CMP  #00         ;
        BEQ  DISPLAY     ;
        AND  #%00001111;
        TAY              ;
        DEY              ;
        CPY  #$FF        ;
        BEQ  DOWN1       ;
        STY  U1          ;
        LDA  U7          ;
        AND  #%11110000;
        CLC              ;
        ADC  U1          ;
        STA  U7          ;
        JMP  DISPLAY     ;
DOWN1   LDA  U7          ;
        AND  #%11110000;
        SEC              ;
        SBC  #16         ;
        CLC              ;
        ADC  #9          ;
        STA  U7          ;
        JMP  DISPLAY     ;

;**********   ARRANGE SONG   **********

SORT4   LDX  #0          ;BUTTON ARRANGE
SORT8   LDA  KEY         ;
        CMP  D.2,X       ;
        BEQ  SORT9       ;
        INX              ;
        CPX  #8          ;
        BNE  SORT8       ;
        JMP  SORT7       ;
SORT9   INX              ;
        TXA              ;
        LDX  U13         ;
        DEX              ;
        STA  BUT,X       ;
        STA  $0771,X     ;
SORT7   LDX  #40         ;WAIT
SORT11  LDY  #00         ;
SORT10  DEY              ;
        BNE  SORT10      ;
        DEX              ;
        BNE  SORT11      ;
        LDA  KEY         ;
        CMP  #43         ;* ?
        BNE  SORT12      ;
        LDA  #31         ;
        LDX  U13         ;
        DEX              ;
        STA  TOP,X       ;
        STA  $0749,X     ;
        LDA  #32         ;
        STA  BUT,X       ;
        STA  $0771,X     ;
SORT12  LDA  KEY         ;
        CMP  #60         ;SPACE ?
        BNE  SORT13      ;
        LDA  #32         ;
        LDX  U13         ;
        DEX              ;
        STA  TOP,X       ;
        STA  $0749,X     ;
        STA  BUT,X       ;
        STA  $0771,X     ;

SORT13  JMP  SORT        ;

;**********   PLAY PATTLINE   **********

PLAY    LDA  #0          ;
        LDX  U5          ;
PLAY2   CPX  #1          ;
        BEQ  PLAY1       ;

PLAY10  CMP  #45         ;
        BNE  PLAY11      ;
        JSR  COW         ;

PLAY11  INC  U10         ;
        INC  U1          ;

        LDY  U7          ;SPEEDWAIT
PLAY14  BEQ  PLAY13      ;
        LDX  #00         ;
PLAY12  DEX              ;
        BNE  PLAY12      ;
        DEY              ;
        JMP  PLAY14      ;

PLAY13  LDA  U10         ;
        CMP  #32         ;
        BEQ  PLAY15      ;
        LDA  KEY         ;
        CMP  #63         ;
        BEQ  PLAY15      ;
        JMP  PLAY16      ;

PLAY15  RTS              ;

;**********   PLAY PATTLINE   **********

DOIT    LDA  SHIFT       ;
        BEQ  DOIT1       ;
        LDA  V+17        ;
        AND  #%11101111;
        STA  V+17        ;
DOIT1   LDA  U5          ;
        STA  G02         ;
        LDA  U8          ;
        STA  G03         ;
        LDA  U9          ;
        STA  G04         ;
D04     LDA  #0          ;
        STA  G01         ;

REPEAT  LDX  G01         ;
        LDA  $0771,X     ;
        TAX              ;
        DEX              ;
        LDA  D.BOLO,X    ;
        STA  U8          ;
        LDA  D.BOHI,X    ;
        STA  U9          ;
```

```
SORT    LDA  KEY        ;
        CMP  #2         ;
        BNE  SORT1      ;
        LDA  SHIFT      ;
        BEQ  SORT2      ;
        LDA  U13        ;MOVE LEFT
        CMP  #1         ;
        BEQ  SORT1      ;
        DEC  U13        ;
        LDA  V+2        ;
        SEC             ;
        SBC  #8         ;
        STA  V+2        ;
        BCS  SORT1      ;
        LDA  MSB        ;
        AND  #%11111101;
        STA  MSB        ;
        JMP  SORT1      ;
SORT2   LDA  U13        ;MOVE RIGHT
        CMP  #38        ;
        BEQ  SORT1      ;
        INC  U13        ;
        LDA  V+2        ;
        CLC             ;
        ADC  #8         ;
        STA  V+2        ;
        BCC  SORT1      ;
        LDA  MSB        ;
        ORA  #%00000010;
        STA  MSB        ;
SORT1   LDA  KEY        ;
        CMP  #63        ;STOP ?
        BNE  SORT3      ;
        JMP  END        ;
SORT3   LDX  #00        ;TOP ARRANGE
SORT6   LDA  KEY        ;
        CMP  D.1,X      ;
        BEQ  SORT5      ;
        INX             ;
        CPX  #8         ;
        BNE  SORT6      ;

        JMP  SORT4      ;
SORT5   TXA             ;
        CLC             ;
        ADC  #49        ;
        LDX  U13        ;
        DEX             ;
        STA  TOP,X      ;
        STA  $0749,X    ;

        CLC             ;
        ADC  #32        ;
        DEX             ;
        JMP  PLAY2      ;
PLAY1   STA  U1         ;
        LDA  #0         ;
        STA  U10        ;
PLAY16  LDY  U1         ;
        LDA  (U8),Y     ;

        CMP  #37        ;COMPUTER DRUMS
        BNE  PLAY3      ;
        JSR  SPACE      ;
        JMP  PLAY11     ;

PLAY3   CMP  #38        ;
        BNE  PLAY4      ;
        JSR  BD         ;
        JMP  PLAY11     ;

PLAY4   CMP  #39        ;
        BNE  PLAY5      ;
        JSR  SD         ;
        JMP  PLAY11     ;
PLAY5   CMP  #40        ;
        BNE  PLAY6      ;
        JSR  CYM        ;
        JMP  PLAY11     ;

PLAY6   CMP  #41        ;
        BNE  PLAY7      ;
        JSR  HH         ;
        JMP  PLAY11     ;

PLAY7   CMP  #42        ;
        BNE  PLAY8      ;
        JSR  WOOD       ;
        JMP  PLAY11     ;

PLAY8   CMP  #43        ;
        BNE  PLAY9      ;
        JSR  TOMHI      ;
        JMP  PLAY11     ;

PLAY9   CMP  #44        ;
        BNE  PLAY10     ;
        JSR  TOMLO      ;
        JMP  PLAY11     ;

        LDX  G01        ;
        LDA  $0749,X    ;
        CMP  #31        ;
        BNE  D02        ;
        JMP  D04        ;
D02     SEC             ;
        SBC  #48        ;
        STA  U5         ;
        JSR  PLAY       ;
        LDA  KEY        ;
        CMP  #63        ;
        BNE  D01        ;
DOEND   LDA  G02        ;
        STA  U5         ;
        LDA  G03        ;
        STA  U8         ;
        LDA  G04        ;
        STA  U9         ;
        JMP  END        ;
D01     INC  G01        ;
        LDA  G01        ;
        CMP  #37        ;
        BNE  REPEAT     ;
        JMP  D04        ;

;**********   LOAD AND SAVE   **********

STORA   LDA  #0         ;
        STA  U1         ;
        JSR  CLEAR      ;WRITE 'LOAD OR
        LDX  #$00       ;        SAVE'
STORA1  LDA  D.LOSA,X   ;
        STA  LINE,X     ;
        INX             ;
        CPX  #12        ;
        BNE  STORA1     ;

STORA4  LDA  KEY        ;
        CMP  #63        ;
        BNE  STORA2     ;
        JSR  CLEAR      ;
        JMP  END        ;
STORA2  CMP  #42        ;
        BNE  STORA3     ;
        LDA  #0         ;LOAD SELECTED
        STA  U1         ;
        JMP  NAME       ;
STORA3  CMP  #13        ;
        BNE  STORA4     ;
```

```
         LDA  #1          ;SAVE SELECTED
         STA  U1          ;
         JMP  NAME        ;

CLEAR    LDA  #32         ;CLEAR DISPLAY-

         LDX  #0          ;LINE
CLEAR1   STA  LINE,X      ;
         INX              ;
         CPX  #23         ;
         BNE  CLEAR1      ;
         RTS              ;

NAME     JSR  WAIT        ;
         JSR  WAIT        ;
         JSR  CLEAR       ;
         LDA  #0          ;
         STA  U11         ;
         LDA  #64         ;
         STA  U10         ;

NAME3    JSR  WAIT        ;INPUT NAME
NAME6    LDA  KEY         ;
         CMP  U10         ;
         BEQ  NAME6       ;
         STA  U10         ;
         LDX  #0          ;
NAME2    LDA  KEY         ;
         CMP  D.NAME,X    ;
         BEQ  NAME1       ;
         INX              ;
         CPX  #37         ;
         BNE  NAME2       ;
         LDA  KEY         ;
         CMP  #63         ;STOP ?
         BNE  NAME4       ;
         JSR  CLEAR       ;
         JMP  END         ;
NAME4    CMP  #0          ;DEL ?
         BNE  NAME5       ;
         LDX  U11         ;
         LDA  #32         ;
         STA  LINE,X      ;
         CPX  #0          ;
         BEQ  NAME3       ;
         DEC  U11         ;
         LDX  U11         ;
         STA  LINE,X      ;
         JSR  WAIT        ;
         JMP  NAME3       ;
NAME5    CMP  #1          ;RETURN ?
         BNE  NAME3       ;
         JMP  OPERATE     ;
NAME1    LDA  D.POKE,X    ;
         LDX  U11         ;
         STA  LINE,X      ;
         CPX  #7          ;
         BEQ  NAME3       ;
         INC  U11         ;
         JMP  NAME3       ;

OPERATE  LDX  #0          ;SET UP

OPERATE1 LDA  LINE,X      ;FILENAME
         STA  D.FILE,X    ;
         INX              ;
         CPX  #8          ;
         BNE  OPERATE1    ;
         LDY  #0          ;
OPERATE2 LDA  D.FILE,Y    ;
         CMP  #30         ;
         BCS  OPERATE3    ;
         CLC              ;
         ADC  #64         ;
         STA  D.FILE,Y    ;
OPERATE3 INY              ;
         CPY  #8          ;
         BNE  OPERATE2    ;
         LDA  U1          ;
         CMP  #1          ;
         BNE  OPERATE4    ;
         LDA  #<D.START   ;DISK-SAVE
         STA  GO1         ;
         LDA  #>D.START   ;
         STA  GO2         ;
         LDA  #8          ;
         JSR  FILEPARA    ;
         LDA  #18         ;
         LDX  #<D.OVER    ;
         LDY  #>D.OVER    ;
         JSR  FILENAME    ;
         LDA  #GO1        ;
         LDX  #<D.END     ;
         LDY  #>D.END     ;
         JSR  SAVE        ;
         JMP  END         ;

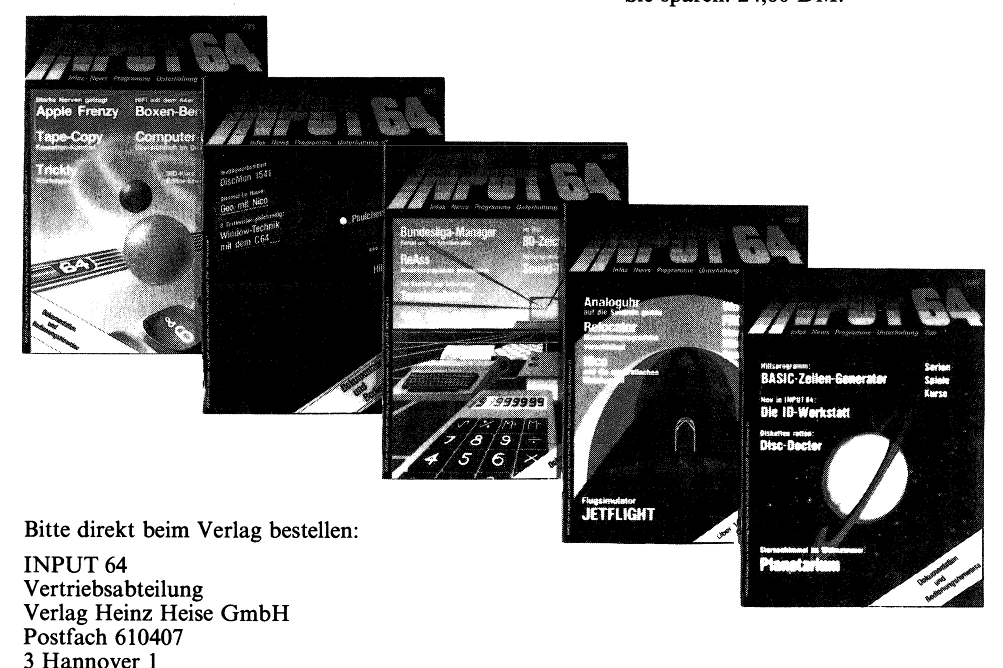
OPERATE4 LDA  #8          ;DISK-LOAD
         LDY  #1          ;
         JSR  FILEPARA    ;
         LDA  #16         ;
         LDX  #<D.LINE    ;
         LDY  #>D.LINE    ;
         JSR  FILENAME    ;
         LDA  #0          ;
         JSR  LOAD        ;
         JSR  SETUP       ;
         JSR  DISPLAY     ;
         LDX  #0          ;
OPERATE5 LDA  TOP,X       ;
         STA  $0749,X     ;
         LDA  BUT,X       ;
         STA  $0771,X     ;
         INX              ;
         CPX  #38         ;
         BNE  OPERATE5    ;

         JMP  END         ;
```

# Für Nachzügler und "Spätzünder"

Wegen der großen Nachfrage haben wir bereits vergriffene Ausgaben von INPUT 64 nachproduziert, so daß alle bisher erschienenen Ausgaben wieder lieferbar sind. Ab Ausgabe 4/85 ist INPUT 64 auch auf Diskette erhältlich. Preis: Kassettenversion 12,80 DM / Diskettenversion 19,80 DM. (jeweils incl. Porto und Verpackung)

Außerdem können Sie die Ausgaben 4/85 bis 9/85 der Diskettenversion im Sechser-Pack beziehen. Komplettpreis: 90 DM. Sie sparen: 24,80 DM!




Bitte direkt beim Verlag bestellen:

INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 610407
3 Hannover 1

(Lieferung nur gegen Vorkasse, bitte Verrechnungsscheck beilegen)

## Aus dem Inhalt

Ausgabe 1 – Dateiverwaltung, drei (!) Spiele * Ausgabe 2 – Textprogramm, Zeichensatzeditor * Ausgabe 3 – Spriteeditor, Maschinensprache-Monitor * Ausgabe 4 – SuperTape D II, Grafikerweiterung, Urlaubskalender * Ausgabe 5 – Mathe mit Nico (Teil 1), Talk to me (Dialogsimulation), Hintergrundmonitor * Ausgabe 6 – Textadventure "Fuchsjagd", SID-Kurs (Teil 1), Recorder-Justage, BASIC-Compactor * Ausgabe 7 – HiFi-Boxen-Berechnung, TapeCopy, Scroll Editor * Ausgabe 8 – Discmonitor, Reisekostenberechnung, Musik-Harware * Ausgabe 9 – Reassembler, Reisekostenberechnung, Bundesliga-Simulation * Ausgabe 10 – Flugsimulator, Maschinensprache-Relocator * Ausgabe 11 – Planetarium, Datei-Kopierprogramm, DiscDoctor *

# 3000 Mark

# warten auf den Gewinner!

**Der Wettbewerb geht weiter.**

**Hier noch einmal kurz die Bedingungen:**
**Sie können einsenden:**

- **Grafikprogramme**
- **Musikprogramme**
- **Spiele-**
- **Lernprogramme**
- **Anwenderprogramme**

**und natürlich völlig neue Programmideen.**

**Wichtig: Werfen Sie einen Blick in das Kapitel "Hinweise für Autoren", damit Ihr Programm auch innerhalb von INPUT 64 lauffähig ist.**

**Der Rechtsweg ist wie immer ausgeschlossen.**

# Nutzen-Kosten-Analyse

Das letzte Wort haben natürlich immer noch Sie. Dieses Programm nimmt Ihnen, um es ganz deutlich zu sagen, notwendige Entscheidungen nicht ab. Es liefert aber nach Vorgaben, die Sie eingeben, eine klare Gegenüberstellung der Kosten und Nutzen verschiedener Alternativen.

Und es ist nicht eingeschränkt auf materielle, also Kaufentscheidungen. Sie können damit genauso gut verschiedene Ehepartner, Ihre Arbeitskollegen oder mögliche Mitreisende für den nächsten Urlaub vergleichen lassen. Spielen wir letzteren Fall einmal durch.

Das Eingabe-Menue verlangt zunächst die Festlegung der Varianten. Die maximale Länge der Namen ist auf drei Zeichen festgelegt. Jedes Feld wird mit RETURN abgeschlossen. Sie geben also ein: ART für Arthur, ERN für Erna und so weiter, falsche Eingaben können Sie mit der DEL-Taste korrigieren. Sechs Varianten können verglichen werden. Sind es weniger, schließen Sie ein leeres Feld mit RETURN ab. Nun geht es an das Festlegen der Kriterien. Die Maßstäbe könnten Großzügigkeit (gibt der künftige Mitfahrer auch mal einen aus?), Autokenntnisse für den Fall einer Panne und Kommunikationsfreudigkeit sein. Jeweils die ersten sieben Zeichen dieser Kriterien – es können auch weniger sein – geben Sie jetzt in die linke Spalte ein. Jedes Feld wird wieder mit RETURN abgeschlossen und gegebenenfalls mit der DEL-Taste korrigiert. Die möglichen zehn Kriterien, die das Programm verarbeitet, werden erfahrungsgemäß nie ausgenutzt, die Spalte "Kriterien" können Sie durch die Eingabe eines Leerfeldes verlassen. Nun sind die Kosten dran, da es sich um eine immaterielle Auswertung handeln soll, geben Sie überall eine "1" ein.

In der jetzt folgenden Spalte "Gewichtung" schlägt natürlich der subjektive Faktor voll zu. Ist Ihnen beispielsweise die Kommunikationsfreudigkeit absolut wichtig, geben Sie eine "1" ein. Die Skala der Gewichtung reicht von "1" gleich: absolute Priorität bis "6" wie: wenig wichtig. In diesen Feldern springen die beiden senkrechten Balken, die das aktuelle Eingabefeld markieren, nach jeder Eingabe ein Feld weiter. Zurück, nach links oder auf- und abwärts bewegen Sie es mit den Cursor-Tasten.

Ebenso subjektiv sind die nun vorzunehmenden Bewertungen. Ist Arthur ein Geizkragen, bekommt er in der Reihe "Großzügigkeit" eine "6", die spendable Erna eine "1". Ist alles korrekt eingetippt, können Sie sich mit F1 die Auswertung ansehen. Fehlt eine Angabe, setzt es eine Fehlermeldung.

Die Auswertung erfolgt in Form einer Säulengraphik und der Angabe von Kosten-Nutzen-Punkten. Der jeweilige Punktbeste ist gelb markiert. Wollen Sie Ihre Bewertungen in der Eingabemaske korrigieren, kommen Sie mit F1 zurück. Sie können prinzipiell sämtliche Felder verändern. Die invers unterlegten müssen Sie immer entweder bestätigen (RETURN) oder bei Veränderungen das jeweilige Feld vorher löschen (DEL-Taste). Einen kompletten Neustart des Programms erreichen Sie durch Betätigen der Taste F8.

Gilt es, eine Kaufentscheidung durch den Computer untermauern zu lassen, müssen auch die Kosten sinvoll eingegeben werden. Die Begrenzung auf drei Ziffern erfordert, daß bei Beträgen über 999 DM nur die führenden drei Ziffern eingegeben werden. Dabei ist darauf zu achten, daß die gewählte Einheit beibehalten wird. Wurde zum Beispiel für 195.000 DM 195 eingegeben, entspricht dies für 80.000 DM der Ziffernfolge 080. Die Auswertung bezieht den Preis mit ein, es heißt dann "Nutzen-Punkte je DM".

Wenn Sie Wert auf schriftliche Fixierung der Ergebnisse legen, sollten Sie als 64er-Besitzer vorher die in dieser Ausgabe veröffentlichte Hardcopy-Routine geladen und initialisiert haben. Die Programm-Version für den PC128 hat eine eigene Hardcopy-Funktion.

Womit wir zu einer Neuerung kämen: Nach dem Laden des Moduls Nutzen-Kosten-Analyse aus dem Magazin heraus können Sie wählen zwischen der 128er und der 64er-Version dieses Programms! Die Fassung für den PC128 läuft natürlich nicht innerhalb des Magazins, Sie können sie aber auf Ihren eigenen Datenträger durch CTRL und s überspielen und von dort in Ihren 128er laden. Es arbeitet dann – durch gelgentliche Verwendung des FAST-Modus, also des 2 MHertz-Taktes - um einiges schneller. Außerdem sind die Druckroutinen im Programm enthalten.

Die 64er-Version können Sie auch innerhalb des Magazins ausprobieren und – natürlich – ebenfalls auf Ihre eigene Kassette oder Diskette abspeichern. Haben Sie die 64er-Version einmal gestartet, kommen Sie durch Neustart des gesamten Moduls (CTRL/i und RETURN) zurück ins Auswahl-Menue.

# Listing Nutzen-Kosten-Analyse

```
10 printchr$(147)chr$(142)chr$(8):m0$=chr$(99):sp$=chr$(32):sh$=chr$(160)
20 cr$=chr$(29):cu$=chr$(145):cd$=chr$(17):cl$=chr$(157):hm$=chr$(19)
30 ch$=chr$(147):de$=chr$(20):pu$=chr$(46):ze$=chr$(48):g9$=chr$(98)
40 ro$=chr$(18):nr$=chr$(146):ye$=chr$(158):re$=chr$(28):gr$=chr$(30)
50 bl$=chr$(31):el$=chr$(169):er$=chr$(223)
60 rem** nutzen-kosten-analyse **
70 rem**          by            **
80 rem** w. zott  & e. kuegler **
90 rem**       sept. 85         **
100 m1$=m0$:m1$=m1$+m1$+m1$+chr$(178)+m1$+m1$+m1$+chr$(178)
110 m2$=g9$+sp$+sp$+sp$:m2$=m2$+m2$
120 m3$=m0$+m0$+chr$(123):m3$=m3$+m3$
130 m4$=left$(m3$,3)+chr$(177):m4$=m4$+m4$
135 m5$=chr$(192):m5$=m5$+m5$:m5$=m5$+m5$
137 ml$=sp$+sp$+ro$+sp$+sp$
140 a$=chr$(144)+sh$+sh$+sh$+sh$+sh$+ye$
150 a$=a$+"nutzen   kosten   analyse"+sh$
160 a$=a$+sh$+sh$+sh$
170 printhm$;:fori=1to8:printcd$:next
180 fori=1to34:printcr$;:next
190 getja$:ifja$=chr$(133)then460
200 poke53280,0:poke53281,0
210 fori=1to36:printleft$(a$,i);:forj=1toi-1:printcl$;:next:next
220 printhm$chr$(154):print:print
230 printml$er$nr$spc(5)ro$sp$sp$nr$spc(6)ro$sp$sp$nr$spc(3);
231 printro$el$sp$nr$el$spc(8)ro$el$er$
240 fori=1to8:printcd$;:next:gosub420:goto 400
250 printml$sp$er$nr$spc(4)ro$sp$sp$nr$spc(6)ro$sp$sp$nr$sp$sp$ro$el$;
251 printsp$nr$el$spc(8)ro$el$sp$sp$er$
260 fori=1to6:printcd$;:next:gosub420:goto380
270 printml$nr$er$ro$sp$er$nr$sp$ml$nr$spc(4)ml$nr$sp$ro$el$sp$nr$el$;
271 printspc(8)ro$el$sp$nr$el$er$ro$sp$er$
280 printcd$cd$cd$:gosub420:goto360
290 printml$nr$sp$er$ro$sp$er$nr$ml$nr$spc(4)ml$el$sp$nr$el$spc(8);
291 printro$el$sp$nr$el$sp$sp$er$ro$sp$er$
300 printcd$:gosub420:goto340
310 printml$nr$ml$nr$ml$nr$spc(4)ml$sp$nr$el$spc(7)ml$nr$sp$sp$ml$nr$
320 printml$nr$ml$nr$ml$nr$spc(4)ml$sp$er$nr$spc(7)ml$nr$sp$sp$ml$nr$
330 goto 430
340 printml$nr$sp$sp$er$ro$sp$er$nr$sp$ro$sp$sp$nr$spc(4)ml$nr$er$ro$sp$er$;
341 printnr$spc(6)ml$sp$sp$sp$sp$sp$sp$nr$
350 fori=1to3:printcu$;:next:gosub420:goto310
360 printml$nr$spc(3)er$ro$sp$er$sp$sp$nr$spc(4)ml$nr$sp$er$ro$sp$er$nr$;
361 printspc(5)ml$sp$sp$sp$sp$sp$sp$nr$
370 fori=1to5:printcu$;:next:gosub420:goto290
380 printml$nr$spc(4)er$ro$sp$sp$sp$nr$spc(4)ml$nr$sp$sp$er$ro$sp$er$nr$;
381 printspc(4)ml$nr$sp$sp$ml$nr$
390 fori=1to7:printcu$;:next:gosub420:goto270
400 printml$nr$spc(5)er$ro$sp$sp$nr$spc(4)ml$nr$spc(3)er$;
401 printro$sp$er$nr$spc(3)ml$nr$sp$sp$ml$nr$
410 fori=1to9:printcu$;:next:goto250
420 fori=1to10:next:return
430 rem
440 printbl$;:fori=1to13:printcd$;:next:print"by w. zott  &  e. kuegler   v 2.4
 1985
450 gosub3610
460 rem variablendef
470 dimkr$(10),k$(15),kl$(15),f$(15,40),be%(6,10),kp%(10),np(6,10),bh%(6,10)
480 l$=bl$+g9$+cr$+cr$+cr$+g9$
490 kr$(1)=chr$(28 )+"preis"
500 c$=chr$(5)+g9$+cr$+cr$+cr$+g9$:x=2:y=14
510 rem bildaufbau
520 ifin=0thenprintbl$ch$;:goto540
530 printbl$hm$;
540 poke53280,0:poke53281,0
1250 iflen(kr$(j))<7thenkr$(j)=kr$(j)+sp$:goto1250
1260 ifleft$(kr$(10),1)=sp$thenkr$(10)=""
1270 printkr$(j)
1280 gosub1640:printkl$(x);
1290 x=x+1:gosub1640
1300 ifj<10then printk$(x)cl$;
1310 nextj
1320 ifkz<>0thenreturn
1330 rem sprung preis  feld
1340 a$="":x=4:y=14:gosub1640:printc$;:printcl$;
1350 forj=1to6
1360 poke198,0
1370 gosub3610
1380 ifa$=de$andf$(4,y)<>""thenf$(4,y)=left$(f$(4,y),len(f$(4,y))-1):gosub1890
1390 ifa$=chr$(13)then 1460
1400 ifa$<ze$ora$>chr$(57)then1360
1410 f$(x,y)=f$(x,y)+a$
1420 iflen(f$(x,y))>3thenf$(x,y)=left$(f$(x,y),3):gosub1820
1430 ifkz=1thenprintro$;
1440 fori=1tolen(f$(x,y)):printcl$;:next:printye$re$f$(x,y)nr$;
1450 goto 1360
1460 ifval(f$(x,y))<1thengosub1820:gosub1890:f$(x,y)="":goto1360
1470 fori=1to4:printcl$;:next:printl$;
1480 ifva$(j+1)=""thenj=6
1490 ifj<6then printcl$;:printc$;
1500 printcl$;:y=y+4:nextj
1510 rem sprung eingabafeld
1520 x=5:y=10:gosub1640:printc$;
1530 gosub3610
1540 ifa$=chr$(133)thenprintch$:goto2390.
1550 ifa$=cr$ora$=cd$ora$=cu$ora$=cl$thengosub1710:goto1530
1560 ifa$<chr$(49)ora$>chr$(54)then1530
1570 f$(x,y)=a$
1580 printcl$cl$cl$;
1590 ify<14thenprintgr$a$;
1600 ify>13thenprintye$a$;
1610 printcr$cr$;
1620 ify=10thena$=cd$:gosub1710:goto1530
1630 ify>10thena$=cr$:gosub1710:goto1530
1640 printhm$;
1650 k$(x)=chr$(5)+g9$+cr$+cr$+cr$+cr$+cr$
1660 k$(x)=k$(x)+cr$+cr$+g9$
1670 kl$(x)=chr$(144)+sh$+cr$+cr$+cr$+cr$+cr$
1680 kl$(x)=kl$(x)+cr$+cr$+chr$(144)+sh$
1690 fori=1tox:printcd$;:next
1700 fori=1toy:printcr$;:next:return
1710 printcl$cl$cl$cl$cl$   ;:printl$;
1720 ifa$=cr$then y=y+4
1730 ifa$=cd$then x=x+1
1740 ifa$=cu$then x=x-1
1750 ifa$=cl$then y=y-4
1760 ifx<5 thenx=5
1770 ify<10theny=10
1780 ify>34theny=14:x=x+1
1790 ifx>13thenx=5:y=y+4
1800 gosub 2000
1810 gosub1640:printc$;:return
1820 poke53280,2:fori=1to99:next:poke53280,0:return
1830 rem korrek.kl.feld oben
1840 fori=1to3:printcl$;:next:print"    ";
1850 if kz=1thenprintro$;
1860 fori=0tolen(va$(j))-1 :printcl$;:next:printye$va$(j)nr$;
1870 ifva$(j)=""thenprintcr$;
1880 return
1890 rem korrek.preis feld
1900 fori=1to3:printcl$;:next:print"    ";
1910 if kz=1thenprintro$;
1920 fori=0tolen(f$(4,y))-1 :printcl$;:next:printre$f$(4,y)nr$;
1930 iff$(4,y)=""thenprintcr$;
1940 return
```

```
550 printspc(14)chr$(176)m5$chr$(192)ye$" varianten "bl$m5$left$(m5$,3)chr$(174)
560 fori=1to10:printm0$;:next:print    chr$(178);
570 printm1$m1$m1$;
580 printm0$m0$m0$chr$(174)
590 printye$"kriterien "bl$;:printg9$;:printgr$"gew"bl$;
600 printm2$m2$m2$;
610 printg9$
620 fori=1to10:printm0$;:next:printchr$(123);
630 printm3$m3$m3$;
640 printm0$m0$m0$chr$(179)
650 forj=1to10:gosub660:next:goto690
660 fori=1to10:printsp$;:next:printg9$sp$sp$sp$;
670 printm2$m2$m2$;
680 printg9$:return
690 fori=1to10:printm0$;:next:print    chr$(177);
700 printm4$m4$m4$;
710 printm0$m0$m0$chr$(189)
720 printhm$cd$cd$cd$cd$cr$cr$kr$(1)
730 fori=1to5:printcd$:next:printgr$"gewichtung:";
740 printre$"                bewertung:"
750 printchr$(154)"abs.prioritaet=1          sehr gut      =1
760 print"vorranig      =2         gut          =2"
770 print"sehr wichtig  =3         befriedigend =3"
780 print"wichtig       =4         ausreichend  =4"
790 print"beachtenswert =5         mangelhaft   =5"
800 print"wenig.wichtig =6         ungenueg.    =6"
810 fori=1to39:printbl$chr$(183);:next
820 printchr$(151)" f1 => berechnung          f8 => neustart"bl$
830 printro$"  by   e. kuegler   &   w. zott   1985   "nr$cu$;
840 ifkz>0thengosub2160
850 ifkz<>0then gosub 890
860 ifkz>0andik<10thengosub1080
870 ifkz<>0thengosub 1330
880 ifkz>0then1510
890 gosub1640:printc$;:printcl$;
900 rem 1.zeile
910 forj=1to6
920 gosub3610
930 ifa$=de$andva$(j)<>""thenva$(j)=left$(va$(j),len(va$(j))-1):gosub1830
940 ifa$=chr$(13)andva$(1)<>""then1010
950 ifa$<ze$ora$>chr$(90)then920
960 va$(j)=va$(j)+a$
970 iflen(va$(j))>3thenva$(j)=left$(va$(j),3):gosub1820
980 ifkz=1thenprintro$;
990 fori=1tolen(va$(j)):printcl$;:next:printye$va$(j)nr$;
1000 goto 920
1010 fori=1to4:printcl$;:next:printl$;
1020 ifva$(j)=""thenj=6
1030 ifj<6then printcl$;:printc$;
1040 printcl$;:nextj
1050 ifkz<>0thenreturn
1060 rem sprung in gr.feld
1070 :
1080 x=4:y=2:gosub1640:printkr$(1);
1090 ifkz=0thenx=5:y=1:gosub1640:printk$(x);:printcl$;
1100 rem  ********** bewegung im kri.feld
1110 ifkz<>0thenx=ik+4:y=1:gosub1640:printk$(x);:printcl$;
1120 ifkz=0thenforj=2to10
1130 ifkz<>0thenforj=ik+1to10
1140 gosub3610
1150 ifa$=de$andkr$(j)<>""thenkr$(j)=left$(kr$(j),len(kr$(j))-1):gosub1950
1160 rem
1170 ifa$=chr$(13)andkr$(2)<>""then1230
1180 ifa$<ze$ora$>chr$(90)then1140
1190 kr$(j)=kr$(j)+a$
1200 iflen(kr$(j))>7thenkr$(j)=left$(kr$(j),7):gosub1820:goto1140
1210 fori=1tolen(kr$(j)):printcl$;:next:printye$kr$(j);
1220 goto 1140
1230 fori=len(kr$(j))to6+len(kr$(j)):printcl$;:next
1240 ifkr$(j)=""thenj=10
1950 rem korrek. gr.feld
1960 fori=1to7:printcl$;:next:print"         ";
1970 fori=0tolen(kr$(j))-1 :printcl$;:next:printye$kr$(j);
1980 ifkr$(j)=""thenprintcr$;
1990 return
2000 ifva$(1)=""andy>10theny=10
2010 ifva$(2)=""andy>14theny=14:ifx<13 then  x=x+1
2020 ifva$(3)=""andy>18theny=14:ifx<13 then  x=x+1
2030 ifva$(4)=""andy>22theny=14:ifx<13 then  x=x+1
2040 ifva$(5)=""andy>26theny=14:ifx<13 then  x=x+1
2050 ifva$(6)=""andy>30theny=14:ifx<13 then  x=x+1
2060 ifkr$(2)=""andx>4thenx=5:y=14
2070 ifkr$(3)=""andx>5thenx=5:y=14
2080 ifkr$(4)=""andx>6thenx=5:y=14
2090 ifkr$(5)=""andx>7thenx=5:y=14
2100 ifkr$(6)=""andx>8thenx=5:y=14
2110 ifkr$(7)=""andx>9thenx=5:y=14
2120 ifkr$(8)=""andx>10thenx=5:y=14
2130 ifkr$(9)=""andx>11thenx=5:y=14
2140 ifkr$(10)=""andx>12thenx=5:y=14
2150 return
2160 rem wiederaufruf
2170 printhm$cd$cd$;:fori=1to15:printcr$;:next
2180 fori=1to6
2190 ifva$(i)=""then2210
2200 iflen(va$(i))<3thenva$(i)=va$(i)+sp$:goto2200
2210 printye$ro$va$(i)cr$nr$;:next
2220 printhm$cd$cd$cd$cd$cd$;
2230 fori=2to10
2240 printcr$cr$ye$kr$(i):next
2250 printhm$cd$cd$cd$cd$;:fori=1to15:printcr$;:next
2260 fory=14to34step4
2270 iff$(4,y)=""then2290
2280 iflen(f$(4,y))<3thenf$(4,y)=f$(4,y)      +sp$:goto2280
2290 printre$ro$f$(4,y)cr$      nr$;:next
2300 printhm$;
2310 forx=5toik+4:fory=10to10+4*ivstep4
2320 printhm$;:gosub1640
2330 rem
2340 ify<14thenprintgr$cr$cr$f$(x,y)
2350 ify>13thenprintye$cr$cr$f$(x,y)
2360 nexty:nextx
2370 x=2:y=14:return
2380 printch$:kz=1:goto510
2390 rem
2400 rem auswertung
2410 fori=10to1step-1:ik=i
2420 iflen(kr$(i ))<>0theni=-i
2430 next
2440 rem ik=anz. krit.
2450 fori =6to1step-1:iv=i
2460 iflen(va$(i ))<>0theni=-1
2470 poke53280,2:poke53281,2
2480 next
2490 rem iv=anz. vari.
2500 foria=1toiv:forib=1toik
2510 be%(ia,ib)=val(f$(3+ib,10+ia*4))
2520 ifbe%(ia,ib)=0thenprintch$:gosub3550:ib=ik+2:ia=iv+2
2530 nextib,ia:ifia>iv+1then2380
2550 rem umwandlung der zeichenkette aus matrix: be%=bewertung
2560 rem
2580 rem pruefung auf vollst.
2590 fl=0:forib=2toik:gw%(ib)=val(f$(3+ib,10))
2600 ifgw%(ib)=0thenfl=1
2610 next
2620 iffl1thenprintch$:gosub3550:goto2380
2630 rem gw%= gewichtung
```

Fortsetzung Seite 20

## Fortsetzung von Seite 19

```
2640 foria=1toiv:forib=2toik
2650 bh%(ia,ib)=7-be%(ia,ib)
2660 nextib:nextia
2670 rem bh%=hilfsvar.bewert.
2680 poke53280,12:poke53281,12
2690 rem 1. farbwechsel
2700 gs%=0:mx=0:fori=1to6:sp(i)=0:next
2710 foria=2toik:gh%(ia)=7-gw%(ia):gs%=gs%+gh%(ia):next
2720 mu=100/gs%
2730 foria=2toik:gh(ia)=gh%(ia)*mu:gs%=gs%+gh%(ia):next
2740 rem umrechnung der gewicht.
2750 foria=1toiv:forib=2toik
2760 np(ia,ib)=bh%(ia,ib)*gh(ib)
2770 sp(ia)=sp(ia)+np(ia,ib):nextib,ia
2780 rem nutzenpunktber.
2790 foria=1toiv:kp(ia)=sp(ia)/be%(ia,1):ke(ia)=(int(sp(ia)*100/be%(ia,1)))/10
2800 rem nutzenpunktber.
2810 ma(ia)=kp(ia)
2820 ifia=1thenmx=ma(ia):vm=1:next
2830 ifmx<ma(ia)thenmx=ma(ia):vm=ia
2840 next
2850 foria=1toiv:kp(ia)=kp(ia)/(mx/10):kp%(ia)=int(kp(ia)+0.05)
2860 ifkp%(ia)=0thenkp%(ia)=1
2870 next
2880 rem kostennutzenp.+maximalw.
2890 :
2900 rem ausgabe
2910 poke53280,0:poke53281,0:printch$re$;
2920 foria=1toiv
2930 iflen(va$(ia))<3thenva$(ia)=va$(ia)+sp$:goto2930
2940 next
2950 forbi=1to120:printro$sp$;:next
2960 printhm$cd$spc(3)"a u s w e r t u n g"spc(11)"ek+wz"
2970 printspc(3);:forbi=1to19:printro$chr$(163);:next:printspc(9)" 09 1985
2980 printhm$ro$spc(32)"n  k  a":print
2990 forib=1to5:printspc(5)chr$(125):printspc(5)chr$(179):next:printspc(5)chr$(1
25)
3000 printcu$:forbi=1to5:printsp$;:next:printchr$(173);
3010 forib=1to6:printm0$m0$m0$chr$(178)m0$;:next
3020 forib=1to3:printm0$;:next
3030 printhm$spc(25)sp$sp$sp$sp$;:print
3040 printspc(25)sp$sp$sp$sp$
3050 printhm$;:forib=1to3:print:next:print"100%":print
3060 forib=80to20step-20:print""ib;cl$"%":print:next
3070 printhm$:forib=1to13:print:next:printspc(6);
3080 forib=1toiv:printsp$sp$ye$va$(ib);:next
3090 rem ende aufbau koordi.
3100 cw=3
3110 foria=1toiv:cw=cw+5
3120 foric=1tokp%(ia):printhm$;:forcs=1to11-(kp%(ia)-1):print:nextcs
3130 forch=1toic:print:nextch
3140 printspc(cw);ye$ro$sp$sp$;
3150 ific=1then3210
3160 ific=kp%(ia)thenprintnr$re$chr$(169):goto3190
3170 printro$re$sp$
3180 ifkp%(ia)=1thenprintcl$;:goto3160
3190 nextic:ific=chthen3260
3200 nextic:ific=chthen3260
3210 printcu$cu$cl$nr$chr$(164)chr$(164)cd$;
3220 printcl$cl$cl$chr$(110)cr$ro$re$;
3230 printchr$(169)cd$cl$sp$;
3240 ifch=icthengoto3260
3250 goto3180
3260 nextia
3270 rem ende saeulen
3280 forcd=1to2:printcd$:next:printcu$;
3290 np$="nutzen-punkte         "
3300 foria=1toiv:ifbe%(ia,1)<>1thennp$=np$+cl$+cl$+cl$+cl$+cl$+"je dm"
3310 nextia
3320 foria=1toiv:printchr$(150);
3330 ka(ia)=int(ke(ia)*100)
3340 ke(ia)=ka(ia)/100
3350 ifia=vmthenprintye$;
3360 ifia<>1thenprintsp$;
3370 printva$(ia)" erzielt";
3380 ke$(ia)=str$(ke(ia))
3390 hh$=right$(ke$(ia),2):ifleft$(hh$,1)=pu$thenke$(ia)=ke$(ia)+ze$
3400 hh$=right$(ke$(ia),3)
3410 ifleft$(hh$,1)<>pu$thenke$(ia)=ke$(ia)+pu$+ze$+ze$
3420 iflen(ke$(ia))<5thenke$(ia)=ze$+right$(ke$(ia),3)
3430 iflen(ke$(ia))<8thenke$(ia)=sp$+ke$(ia):goto3430
3440 printke$(ia);
3450 printspc(1)np$;
3460 next
3470 rem ende auswertung
3480 printhm$
3490 fori=1to18:print:next
3500 printchr$(151)ro$"f1 => eingabemaske        f8 => neustart"cu$;
3510 gosub3610
3520 ifa$=chr$(133)thenprintch$:goto2380
3530 goto3510
3540 rem fehlermeldung bei unvollst. angaben
3550 printch$
3560 print:print:print:print"   angaben nicht vollstaendig !!
3570 print:print"   bitte ueberpruefen.
3580 print:printspc(3)ro$"taste"nr$
3590 gosub3610
3600 return
3610 poke198,0
3620 geta$:ifa$=""then3620
3625 ifa$=chr$(140)thenrun
3630 return
ready.
```

# SCROLLO

Mit Vollgas durch den Speicher des C64!

Ein Spiel, das Schnelligkeit und Konzentration erfordert. Sie steuern ein Fahrzeug durch den Speicher-Bereich Ihres Rechners im Bereich von $1000 bis $7FFF (40960-32767). Dabei Sie muessen gegen eine(n) Mitspieler(in) antreten.

Passen Sie auf, daß Sie weder Ihre bisher zurückgelegte Bahn, noch die gegnerische Bahn treffen. Außerdem dürfen Sie nicht aus dem Speicherbereich hinausrasen! Oberhalb liegt nämlich der Zeichensatz dieses Spiels, nach unten prallen Sie mit Wucht vor den Reset-Schutz. Versuchen Sie im Spiel dem anderen den Weg abzuschneiden. Wer am längsten durchhält, hat gewonnen.

Mit dem Feuerknopf Ihres Joysticks oder der " " bzw. "DEL" können Sie die Geschwindigkeit Ihres Fahrzeug verändern. Doch VORSICHT!, je größer das Tempo, desto mehr Treibstoff verbrauchen Sie. Um durchzuhalten, können Sie die kleinen Treibstofftanks "F" auf dem Spielfeld aufsammeln, wodurch Ihr Treibstoffvorrat jedesmal wieder aufgefüllt wird. Der Treibstoffvorrat wird unterhalb des Spielfeldes angezeigt. Im oberen Feld spielt Spieler(in) 1, im unteren Spieler(in) 2. Haben Sie zwei Joysticks, erklärt sich die Steuerung fast von selbst. Rauf, runter, rechts und links steuern die entsprechende Richtung Ihres Fahrzeugs. Der Feuerknopf verändert das Tempo.

Zur Tastaturbedienung:

| Spieler 1 | Spieler 2 |
|---|---|
| W | @ Rauf |
| A | : LINKS |
| S | ; Rechts |
| Z | . Runter |
| Pfeil links | DEL Tempo |

Das Spiel wurde der Idee im Film TRON nachempfunden. Für jedes erreichte BYTE (Punkt im Spielfeld) gibt es einen Punkt, für jeden Treibstofftank 50 Punkte.

Viel Spaß und eine ruhige Hand!

# 64er Tips

Diesmal informieren wir Sie über den Umgang mit Druckern. Es werden die notwendigen Befehle wie:

OPEN CMD PRINT# CLOSE

im Einzelnen erklärt. Bezeichnungen wie logische Filenummer, Deviceadresse, Sekundaeradresse und ASCII-Code sollten nach Lektüre der 64er Tips keine unverständlichen Fremdworte mehr sein.

Wie die einzelnen Befehle im Prinzip funktionieren, können Sie sich in einer kleinen Simulation anschauen. Auch wenn Ihr Drucker um einiges schneller sein sollte, dürfte die Reaktion Ihres Geräts in etwa unserem "Modell" entsprechen.

Die Angebotspalette der Drucker ist kaum noch überschaubar. Neben den bekannten Nadel-Druckern und Typenrad-Maschinen, können auch Plotter oder Tintenstrahldrucker eingesetzt werden. Das Teuerste und zugleich Neuste stellen die Laser-Drucker dar, die jedoch für die meisten C64-Benutzer kaum erschwinglich sein dürften. Im Allgemeinen versuchen die Hersteller die üblichen Standards in der Druckersteuerung einzuhalten. Die Abweichungen werden letztlich von der Intelligenz und den mechanischen Fähigkeiten des jeweiligen Druckertyps bestimmt. So ist der MPS 801 von Commodore nicht in der Lage, seinen Zeilenabstand zu verändern. Dagegen ist das Ausdrucken von HIRES-Grafiken auf dem MPS802 nur auf Umwegen möglich. Wir haben uns im Wesentlichen an den Commodore-Druckern orientiert und uns über die STAR-Drucker, die wir zur Zeit im Test haben, informiert.

Besondere Schwierigkeiten entstehen, wenn ein Drucker mit Centronics-Anschluß über ein Interface an den seriellen Port an den C64 angeschlossen wird. Hier sind die Reaktionen des Drucker in erster Linie von den Fähigkeiten des Zwischengerätes abhängig, des Interfaces, , das für den jeweiligen Drucker die Konvertierung der seriellen BIT-Muster des C64 auf den parallelen Empfangskanal des Druckers vornimmt. Zu einem späteren Zeitpunkt werden wir über die Problematik "Interface" Bericht erstatten.

Grundsätzlich ist eine vernünftige Verwendung des eigenen Druckers (eventuell mit Interface) nur unter Zuhilfenahme des oder der Handbücher möglich, wobei probieren tatsächlich über studieren geht, denn das Papier im Handbuch ist oft geduldiger als das in Ihrem Drucker!

## Hardcopy

Eigentlich gibt es nicht viel an dieser Hardcopy-Routine zu erklären. Sie kennen sie bereits aus dem INPUT 64-Betriebssystem. Ausgedruckt wird der aktuelle Inhalt des Textbildschirms (also nicht Sprites und/oder hochauflösende Graphik!). Aufgerufen wird das Programm wie gewohnt durch die Tastenkombination CTRL und b. Der Rest geschieht automatisch: es wird festgestellt, ob ein Drucker mit Geräteadresse 4, 5 oder 6 angeschlossen ist und, falls vorhanden, auf das entsprechende Gerät gedruckt. Ist kein Drucker angeschlossen, macht das auch nichts weiter. Es gibt weder eine Fehlermeldung noch eine Störung im sonstigen Programmablauf.

Sie können zwischen zwei Versionen des Tools wählen: die eine liegt am BASIC-Anfang. Nach dem Laden vom eigenen Datenträger und dem Start durch RUN ist die Routine initialisiert und kann mit CTRL und b aufgerufen werden. Sie können dann ein BASIC-Programmm nachladen oder neu eingeben. Diese Version hat den Vorteil, daß Sie Ihr Programm samt Hardcopy-Routine abspeichern können. Geben Sie dazu vor dem SAVE-Befehl ein: POKE 44,8. Dadurch wird der BASIC-Anfang wieder heruntergesetzt, und der BASIC-Bereich umfaßt sowohl die Routinen des Tools als auch Ihr eigenes Programm. Wenn Sie ein so abgespeichertes Programm wieder laden und mit RUN starten, initialisiert sich zuerst die Hardcopy-Routine. Anschließend wird Ihr Programm automatisch gestartet.

Die zweite Version wird ebenfalls nach dem Laden mit RUN gestartet, kopiert sich dann aber in einen vom BASIC unabhängigen Adressbereich und initialisiert die Hardcopy-Möglichkeit. Der BASIC-Speicher ist dann frei.

### Programmatisches

Bei der Initialisierung wird (in beiden Versionen) der Vektor auf die Tastaturauswertung ausgelesen, gerettet und dann auf die eigene Abfrage gelegt. Diese prüft auf CTRL und b. Wurden diese Tasten gedrückt, geht's zur Hardcopy-Routine, ansonsten erfolgt ein indirekter Sprung zu der bei der Initialisierung festgestellten Tastaturabfrage. Dies kann die originale Betriebssystem-Routine sein, genausogut aber auch eine selbstgebaute, die vor dem Start des Hardcopy-Programms initialisiert wurde.

Die Version am BASIC-Anfang setzt den BASIC-Start auf $0A01 und beginnt bei $0832. Falls sie durch einen Reset oder durch RUN/STOP und RESTORE 'abgehängt' wurde, kann sie mit einem "SYS 2098" erneut initialisiert werden. Die Version zwischen den ROMs belegt den Bereich ab $C800 und kann gegebenenfalls mit "SYS 52736" neu aktiviert werden. Alles weitere ist dem auf den folgenden Seiten abgedruckten Listing zu entnehmen.

### Drucker-Anpassung

Störend bei den Bildschirmausdrucken ist meist der Zeilenabstand des Druckers. Gerade bei sorgfältig aufgebauten Rahmen gibt dies schwarz auf weiß unschöne Lücken. Bei den meisten Druckern ist dieser Zeilenanstand allerdings einstellbar. Wir haben, weil die Erklärungen in den Drucker-Handbüchern oft etwas dürftig sind, für einige Typen beispielhaft die entsprechenden Sequenzen zusammengestellt.

**VC 1526** möchte diesen Steuerbefehl über Kanal 6 und Sekundäradresse 6 übermittelt bekommen. Der Befehl selbst besteht eigentlich nur aus der Anzahl der Zeilen pro Zoll, und zwar berechnet als Wert=216/(Zeilen pro Zoll). Voreingestellt ist ein Wert von 6 Zeilen pro Zoll, also Wert gleich 36. Um die Leerzeile zu eliminieren, wird gesendet: OPEN6,4,6§RINT#6, CHR$(18):CLOSE6

**MPS 802**siehe VC 1526

Die meisten anderen Drucker realisieren diese Umstellung entweder über eine Einstellung der DIP-Schalter oder über eine entsprechende ESCAPE-Sequenz. ESCAPE ist ein an den Drucker gesendetes Zeichen, das in den Befehls-Modus schaltet. Das heißt, die folgenden Zeichen werden nicht ausgedruckt, sondern als Steuerzeichen ausgewertet. Der ESCAPE-Code ist bei fast allen gängigen Druckern CHR$(27). Ebenfalls für fast alle Drucker gilt, daß das auf das ESCAPE-Signal folgende Zeichen für "Zeilenabstand einstellen" ein "A" beziehungsweise CHR$(65) ist. Die zu übergebende Zahl für den Zeilenabstand selbst variiert je nach Druckertyp, beim Epson FX80 tut's die Sequenz OPEN4,4, 0§RINT#4,CHR$(27)CHR$(65)CHR$(8); Das Semikolon am Ende ist wichtig!

Praktisch ist, mit dieser Druckersteuerung für die Hardcopy-Routine einen Lader zu kombinieren, der das Hardcopy-Programm nachlädt.

# Hilfsprogramm

```
PROFI-ASS 64 V2.0    SEITE 1

1150:  0801                      .OPT P
1150:  0801                                  ;NUR DRUCKEN
1180:  0801                      *=   $0801  ;START (AM BASIC-ANFANG)
1182:  0801 1F 08                .WORDBASEND
1184:  0803 C1 07                .WORD1985
1186:  0805 9E                   .BYTE$9E
1188:  0806 20 32 30             .ASC " 2081 : HARDCOPY INPUT64"
1189:  081E 00                   .BYTE0
1190:  081F 00 00       BASEND   .BYTE0,0
1199:                   ;
1200:                   ;----------------------------------
1210:                   ;  S Y M B O L A D R E S S E N
1220:                   ;----------------------------------
1230:                   ;
1240:                   ;REGISTER
1250:                   ;
1260:  00FD             POINTER  =    $FD     ;ZEIGER AUF BILDSCHIRMZEICHEN
1270:  00CB             TASTE    =    $CB     ;NUMMER DER GEDRUECKTEN TASTE
1280:  028F             ABFRVEK  =    $028F   ;ZEIGT AUF NEUE TASTATURABFRAGE
1290:  D000             VIC      =    $D000
1300:  D400             SID      =    $D400
1310:                   ;
1320:                   ;ROM-ROUTINEN
1330:                   ;
1340:  A7AE             RUN      =    $A7AE   ;INTERPRETERSCHLEIFE
1350:  EB48             ABFRALT  =    $EB48   ;TASTATURABFRAGE IM ROM
1360:  FFBA             SETFLS   =    $FFBA
1370:  FFBD             SETNAM   =    $FFBD
1380:  FFC0             OPEN     =    $FFC0
1390:  FFC3             CLOSE    =    $FFC3
1400:  FFC6             CHKIN    =    $FFC6   ;EINGABE AUF DATEI
1410:  FFC9             CHKOUT   =    $FFC9
1420:  FFCC             CLRCH    =    $FFCC   ;EINGABE AUF STANDART
1430:  FFCF             BASIN    =    $FFCF
1440:  FFD2             BSOUT    =    $FFD2
1450:                   ;
1460:                   ;----------------------------------
1470:                   ;  P R O G R A M M B E G I N N
1480:                   ;----------------------------------
1490:                   ;
1500:                   ;BASIC-ANFANG HOCH
1510:                   ;-----------------
1520:  0821 A9 01                LDA  #$01
1520:  0823 85 2B                STA  $2B
1520:  0825 A9 0A                LDA  #$0A
1520:  0827 85 2C                STA  $2C
1530:  0829 20 59 A6             JSR  $A659   ;CHRGET AUF PROG.-START
1530:  082C 20 32 08             JSR  INIT
1530:  082F 4C AE A7             JMP  RUN
1540:                   ;
1550:                   ;SYSTEMINITIALISIERUNG
1560:                   ;---------------------
1570:  0832 78          INIT     SEI
1580:  0833 AD 8F 02             LDA  ABFRVEK
1580:  0836 AE 90 02             LDX  ABFRVEK+1
1580:  0839 C9 58                CMP  #<ABFRAGE
1580:  083B D0 04                BNE  INIT1
1590:  083D E0 08                CPX  #>ABFRAGE

2110:  08CE A9 00     DR0D     LDA  #$00     ;PLOTTER INIT
2120:  08D0 20 D2 FF           JSR  BSOUT
2120:  08D3 20 CC FF           JSR  CLRCH
2120:  08D6 A9 04              LDA  #4
2120:  08D8 20 C3 FF           JSR  CLOSE
2120:  08DB A9 06              LDA  #$06
2120:  08DD A8                 TAY
2120:  08DE AA                 TAX
2130:  08DF 20 BA FF           JSR  SETFLS
2130:  08E2 20 C0 FF           JSR  OPEN
2130:  08E5 A2 06              LDX  #$06
2130:  08E7 20 C9 FF           JSR  CHKOUT
2140:  08EA AD 18 D0           LDA  VIC+24
2140:  08ED 29 02              AND  #$02
2140:  08EF D0 03              BNE  LOWER
2140:  08F1 A9 30              LDA  #$30
2140:  08F3 2C                 .BYTE$2C
2150:  08F4 A9 31     LOWER    LDA  #$31
2150:  08F6 20 D2 FF           JSR  BSOUT
2150:  08F9 20 CC FF           JSR  CLRCH
2160:  08FC A9 06              LDA  #$06
2170:  08FE 20 C3 FF           JSR  CLOSE
2180:  0901 A9 04              LDA  #$04
2190:  0903 A2 06              LDX  #$06
2200:  0905 A0 00              LDY  #$00
2210:  0907 20 BA FF           JSR  SETFLS
2220:  090A 20 C0 FF           JSR  OPEN
2220:  090D A2 04              LDX  #$04
2220:  090F 20 C9 FF           JSR  CHKOUT
2230:  0912 A9 00     DR0C     LDA  #$00
2240:  0914 85 FD              STA  POINTER
2250:  0916 AD 88 02           LDA  $0288    ;HIGH-BYTE
2260:  0919 85 FE              STA  POINTER+1  ;VIDEO-RAM
2270:  091B 20 ED FF           JSR  $FFED    ;SPALTEN/ZEILEN HOLEN
2280:  091E 8E D5 09           STX  SPALTE
2290:  0921 8C D4 09           STY  ZEILE
2300:  0924 A9 0D              LDA  #$0D     ;CR ALS INIT
2310:  0926 20 D2 FF           JSR  BSOUT
2320:  0929 A0 00              LDY  #$00
2330:  092B AD 18 D0  DRUCK2   LDA  VIC+24   ;ZEICHENSATZ-BASIS
2340:  092E 29 02              AND  #$02     ;KLEINSCHRIFT
2350:  0930 F0 03              BEQ  DRUCK3   ;NEIN
2360:  0932 A9 11              LDA  #$11     ;CURSOR DOWN MODE
2370:  0934 2C                 .BYTE$2C      ;DUMMY (BIT ABSOLUT)
2380:  0935 A9 91     DRUCK3   LDA  #$91     ;CURSOR UP MODE
2390:  0937 20 D2 FF           JSR  BSOUT
2400:  093A B1 FD     DRUCK4   LDA  (POINTER),Y
2410:  093C 8D D6 09           STA  ZEICHEN
2420:  093F 49 80              EOR  #$80
2430:  0941 91 FD              STA  (POINTER),Y
2440:  0943 AD D6 09           LDA  ZEICHEN
2450:  0946 10 09              BPL  DRUCK5   ;NICHT REVERS
2460:  0948 29 7F              AND  #$7F
2470:  094A 48                 PHA
2480:  094B A9 12              LDA  #$12     ;RVS ON
2490:  094D 20 D2 FF           JSR  BSOUT
2500:  0950 68                 PLA
2510:  0951 C9 22     DRUCK5   CMP  #$22
2520:  0953 D0 02              BNE  DRUCK6   ; ' STATT "
2530:  0955 A9 27              LDA  #$27
2540:  0957 C9 20     DRUCK6   CMP  #$20     ;<32 DANN +64
2550:  0959 90 15              BCC  DRUCK7
2560:  095B C9 60              CMP  #$60     ;>95 DANN +64
2570:  095D B0 11              BCS  DRUCK7
2580:  095F C9 40              CMP  #$40
```

```
1590:  083F F0 06                BEQ  INIT2
1600:  0841 8D D2 09 INIT1       STA  BACK
1600:  0844 8E D3 09             STX  BACK+1
1610:  0847 A9 58    INIT2       LDA  #<ABFRAGE   ;SPRUNGVEKTOR FUER TASTATUR-
1620:  0849 8D 8F 02             STA  ABFRVEK     ;ABFRAGE 'UMBIEGEN'
1630:  084C A9 08                LDA  #>ABFRAGE
1640:  084E 8D 90 02             STA  ABFRVEK+1
1650:  0851 A9 00                LDA  #$00
1660:  0853 8D D1 09             STA  FLAG
1670:  0856 58                   CLI
1670:  0857 60                   RTS
1680:                ;
1690:                ;NEUE TASTATURABFRAGE
1700:                ;--------------------
1710:  0858 AD 8D 02 ABFRAGE     LDA  $028D       ;PRUEFE FLAG,
1720:  085B 29 04                AND  #$04        ;OB CTRL GEDRUECKT
1730:  085D D0 03                BNE  PRUEF       ;JA, WEITER PRUEFEN
1740:  085F 6C D2 09 ZURUECK     JMP  (BACK)      ;NEIN, NORMAL WEITER
1750:  0862 A5 CB   PRUEF       LDA  TASTE
1760:  0864 C9 1C                CMP  #$1C        ;CTRL-B
1770:  0866 D0 F7                BNE  ZURUECK
1780:                ;
1790:                ;BILDSCHIRMAUSDRUCK
1800:                ;------------------
1810:  0868 AD D1 09 DRUCKEN     LDA  FLAG
1820:  086B D0 F2                BNE  ZURUECK
1830:  086D AD 11 D0             LDA  VIC+17      ;HIRES
1840:  0870 29 20                AND  #$20
1850:  0872 D0 EB                BNE  ZURUECK     ;JA
1860:  0874 A9 FF                LDA  #$FF
1870:  0876 8D D1 09             STA  FLAG
1880:  0879 A5 FD                LDA  POINTER
1880:  087B 8D CE 09             STA  TEMP
1890:  087E A5 FE                LDA  POINTER+1
1890:  0880 8D CF 09             STA  TEMP+1
1900:  0883 AD 15 D0             LDA  VIC+21      ;SPRITES
1910:  0886 8D D0 09             STA  TEMP+2
1920:  0889 A9 00                LDA  #$00
1930:  088B 8D 15 D0             STA  VIC+21
1940:  088E 20 BD FF             JSR  SETNAM
1950:  0891 A0 00                LDY  #$00        ;SEK.-ADR.
1960:  0893 A2 04                LDX  #$04        ;GERAETEADRESSE
1970:  0895 8A                   TXA              ;FILE-NR.
1980:  0896 20 BA FF             JSR  SETFLS
1990:  0899 20 C0 FF             JSR  OPEN
2000:  089C A2 04                LDX  #$04
2000:  089E 20 C9 FF             JSR  CHKOUT
2010:  08A1 90 12                BCC  DR0C1
2020:  08A3 20 CC FF             JSR  CLRCH
2020:  08A6 A9 04                LDA  #$04
2020:  08A8 20 C3 FF             JSR  CLOSE
2030:  08AB E6 BA                INC  $BA
2030:  08AD 20 C0 FF             JSR  OPEN
2040:  08B0 A2 04                LDX  #$04
2040:  08B2 20 C9 FF             JSR  CHKOUT
2050:  08B5 90 5B    DR0C1       BCC  DR0C
2060:  08B7 20 CC FF             JSR  CLRCH
2060:  08BA A9 04                LDA  #$04
2060:  08BC 20 C3 FF             JSR  CLOSE
2070:  08BF E6 BA                INC  $BA
2070:  08C1 20 C0 FF             JSR  OPEN
2080:  08C4 A2 04                LDX  #$04
2080:  08C6 20 C9 FF             JSR  CHKOUT
2090:  08C9 90 03                BCC  DR0D
2100:  08CB 4C AA 09             JMP  PRTERR      ;KEIN DRUCKER DRAN
```

```
2590:  0961 90 13                BCC  DRUCK9
2600:  0963 A6 BA                LDX  $BA
2600:  0965 E0 06                CPX  #$06
2600:  0967 D0 0A                BNE  DRUCK8
2610:  0969 AD D6 09             LDA  ZEICHEN
2610:  096C 09 80                ORA  #$80
2610:  096E D0 06                BNE  DRUCK9
2620:  0970 18       DRUCK7      CLC
2630:  0971 69 20                ADC  #$20
2640:  0973 18       DRUCK8      CLC
2650:  0974 69 20                ADC  #$20
2660:  0976 20 D2 FF DRUCK9      JSR  BSOUT
2670:  0979 A9 92                LDA  #$92        ;RVS OFF
2680:  097B 20 D2 FF             JSR  BSOUT
2690:  097E A2 FF                LDX  #$FF
2700:  0980 CA       PRTDEL      DEX
2710:  0981 D0 FD                BNE  PRTDEL
2720:  0983 B1 FD                LDA  (POINTER),Y
2730:  0985 49 80                EOR  #$80
2740:  0987 91 FD                STA  (POINTER),Y
2750:  0989 C8                   INY
2760:  098A D0 02                BNE  DRUCKA
2770:  098C E6 FE                INC  POINTER+1
2780:  098E CE D5 09 DRUCKA      DEC  SPALTE
2790:  0991 D0 A7                BNE  DRUCK4
2800:  0993 A9 28                LDA  #$28        ;40 SPALTEN
2810:  0995 8D D5 09             STA  SPALTE
2820:  0998 CE D4 09             DEC  ZEILE
2830:  099B F0 08                BEQ  PRTEND
2840:  099D A9 0D                LDA  #$0D        ;CR
2850:  099F 20 D2 FF             JSR  BSOUT
2860:  09A2 18                   CLC
2870:  09A3 90 86                BCC  DRUCK2
2880:  09A5 A9 0D    PRTEND      LDA  #$0D        ;CR
2890:  09A7 20 D2 FF             JSR  BSOUT
2900:  09AA 20 CC FF PRTERR      JSR  CLRCH
2910:  09AD A9 04                LDA  #$04        ;FILENUMMER
2920:  09AF 20 C3 FF             JSR  CLOSE
2930:  09B2 AD D0 09             LDA  TEMP+2
2940:  09B5 8D 15 D0             STA  VIC+21      ;SPRITES
2950:  09B8 AD CE 09             LDA  TEMP
2950:  09BB 85 FD                STA  POINTER
2960:  09BD AD CF 09             LDA  TEMP+1
2960:  09C0 85 FE                STA  POINTER+1
2970:  09C2 A9 00                LDA  #0
2970:  09C4 85 C6                STA  $C6         ;TASTATURPUFFER LEEREN
2980:  09C6 A9 00                LDA  #$00
2990:  09C8 8D D1 09             STA  FLAG
3000:  09CB 4C 5F 08             JMP  ZURUECK
3010:                ;
3020:                ;--------------------------------
3030:                ;  H I L F S R E G I S T E R
3040:                ;--------------------------------
3050:                ;
3060:  09CE          TEMP        *=   *+3
3070:  09D1 00       FLAG        .BYTE0
3080:  09D2 48 EB    BACK        .BYTE<ABFRALT, >ABFRALT
3090:  09D4 00       ZEILE       .BYTE0
3100:  09D5 00       SPALTE      .BYTE0
3110:  09D6 00       ZEICHEN     .BYTE0
3120:  09D7          BASIC       *=   $0A00
3130:  0A00 00 00 00             .BYTE0,0,0,0
3140:  0A04                      .SYM
U0801-0A04
NO ERRORS
```

## Memory-List

Sie kommen mit diesem Programm im Speicher verborgenen Sprites und Zeichensätzen auf die Spur. Da Memory-List auch unter den ROM's lesen kann, bleibt Ihnen nichts verborgen. Wenn Sie fündig geworden sind, können Sie den betreffenden Speicherbereich auf Ihren eigenen Datenträger sichern.

Wenn Sie Memory-List von Ihrem Datenträger geladen haben, starten Sie das Programm mit RUN und RETURN. Das Programm verschiebt sich jetzt und belegt danach den Speicherbereich $C700 bis $CDF7. Die oberen Bildschirmzeilen dienen zum einen als Ausgabe (für Sprites und Zeichen) und zum anderen als Menueleiste. Der übrige Teil des Bildschirms ist für die direkte Ausgabe des jeweils eingestellten Speicherbereichs reserviert. Diesen Speicherbereich können Sie mit den Funktionstasten (f1 bis f8) verschieben. Wenn Sie einen Speicherbereich (ab der oben links eingeblendeten Adresse) speichern wollen, drücken Sie die S Taste. Sie können (wieder mit den Funktionstasten) das Ende des Speicherbereiches einstellen. Wenn Sie dann alle folgenden Hinweise beachten, wird ein Absolut-File auf Ihren Datenträger geschrieben, das Sie später nur mit 1,1 beziehungsweise 8,1 wieder laden können. Beachten Sie bitte auch, daß Sie keinen Speicherbereich saven können, der unter den ROM's liegt. Dies betrifft den Bereich $A000 - $BFFF (BASIC-Interpreter), $D000 - $DFFF (I/0-Bereich) und $E000 - $FFFF (Betriebssystem).

Wenn Sie beispielsweise einem neuen Zeichensatz auf der Spur sind, können sie mit der Taste 4 einen Speicherblock von 2 kB Länge (mit Beginn des augenblicklichen Zählers) mit dem Bereich ab $3000 tauschen. Schlagartig wird der neue Zeichensatz eingeschaltet und der Bildschirm neu aufgebaut. Diesen Modus koennen Sie mit der SPACE-Taste wieder rückgängig machen. Beachten Sie aber, daß der Zähler weder auf der ZERO-Page noch im I/O-Bereich stehen darf, da sich sonst der Rechner ganz gepflegt aufhängen kann.

Die folgende Tabelle stellt Ihnen alle Funktionen noch einmal übersichtlich vor. Wir wünschen Ihnen eine erfolgreiche Suche in Ihrem 64er Speicher.

### Befehlsübersicht

| Taste | Funktion |
|---|---|
| f1 | scrollt 40 Bytes vor |
| f2 | scrollt 256 Bytes (Page) vor |
| f3 | scrollt 40 Bytes zurück |
| f4 | scrollt 256 Bytes (Page) zurück |
| f5 | scrollt 1 Byte vor |
| f6 | scrollt 64 Bytes (Sprite-Block) vor |
| f7 | scrollt 1 Byte zurück |
| f8 | scrollt 64 Bytes (Sprite-Block) zurück |
| + | Multicolour-Modus einschalten |
| – | Multicolour-Modus ausschalten |
| 1/2/3 | Farbwahl des Mobs |
| 4 | Speicheraustausch (siehe Text) |
| Space | macht Austausch rückgängig |
| S | Aufruf der Save-Funktion |
| Pfeil links | Hardcopy vom Bildschirm |
| Pfeil hoch | Programmausstieg |

Innerhalb von INPUT 64 sind einige Funktionen aus technischen Gründen eingeschränkt. So können Sie beispielsweise nur den Speicherbereich bis $7FFF ansehen, und die Save- und Druckfunktionen sind abgeschaltet.

---

## Physik mit Nico

Nachdem sich Nico in der letzten Folge eine gewisse Zeit lang geschwinde auf den Weg gemacht hat, wird er diesmal mit einer Masse Kraft inpulsiv beschleunigen. Es geht also diesmal um die physikalischen Begriffe Masse, Kraft und Impuls.

Ihnen steht bekanntlich eine Rechenseite zur Verfügung, auf der Sie alle Rechenoperationen ausführen können, die Ihnen vom Direkt-Modus her bekannt sind. Diese Rechenseite können Sie mit der Taste R und RETURN immer dann aufrufen, wenn Sie zu einer Eingabe aufgefordert werden.

# KALAHA

Das bekannte Afrikanische Bohnen-Spiel

Auch wenn das Wetter hier bestimmt mit dem Afrikanischen Klima nicht mithalten kann, kann Ihnen bei diesem Spiel trotzdem warm werden.

Die besten Chancen haben Sie, wenn Sie mit Überlegung und Strategie zuwerke gehen. Das Spiel erfordert mehr strategisches Geschick, als man auf den ersten Blick vermuten sollte. Sie können gegen Ihren Rechner oder einen Partner antreten.

Es geht darum, durch geschicktes Taktieren die meisten Bohnen in seiner "Heimmulde" einzuheimsen. Die unteren Mulden sind die eigenen, die oberen gehören dem Mitspieler oder dem Rechner. Man nimmt aus einer Mulde die Bohnen auf und verteilt sie einzeln in die nachfolgenden Mulden und zwar im Uhrzeigersinn. Erreicht man in der letzten Mulde eine Anzahl von zwei, vier oder sechs Bohnen, so darf man diese in seine Heimmulde legen. Wurden in der Mulde davor ebenfalls zwei, vier oder sechs Bohnen angehäuft, so gewinnt man auch diese. Entsprechend gilt dies auch für die vorvorletzte, etcetera, etcetera, etcetera. Sind die Mulden einer Seite vollständig entleert, ist das Spiel zuende, und der mit den meisten Bohnen "im Topf" gewinnt. In jeder Runde muß gezogen werden, auch wenn es mal nichts zu holen gibt!

Zur Steuerung der Auswahl benutzen Sie entweder einen Joystick oder die beiden Cursor-Tasten. Mit dem Feuerknopf oder der SPACE-Taste können Sie die Bohnen aufnehmen.

Die Verteilung der Bohnen und das Herausnehmen der Gewinne, sowie die Punktabrechnung erledigt der Rechner für Sie.

Viel Vergnügen beim Ausknobel der Strategie!

### Kommunikationsspiel:

## Quiwi

"Angeregt zu diesem Spiel wurden wir durch die Tatsache, daß viele Computer-Besitzer lange Zeit alleine vor ihrem Gerät verbringen, was zu Konflikten mit Partner, Familie usw. führen kann. QUIWI hingegen ist ein Gesellschaftsspiel, welches die Kommunikation fördert, und somit 'das etwas andere Computerspiel' meinen die Hersteller des diesen Monat erschienenen Spiels. QUIWI ist ein Quizspiel mit rund 4000 Fragen aus den Wissensgebieten Allgemeinwissen, Erdkunde, Geschichte und Politik, Kunst und Literatur, Sport und Humor, Wissenschaft und Technik. Der 64er sucht aus der jeweiligen Datei die 'passende' Frage aus und hat auch die richtige Antwort parat. Die Entscheidung, ob die Antwort des Users richtig oder falsch war, treffen hingegen die maximal 14 Mitspieler. Durch die große Anzahl der Fragen dürften Wiederholungen, die solch ein Spiel langweilig werden lassen, auch kaum vorkommen. Schon die uns vorliegende Demo-Version mit nur 360 Fragen hinterließ einen guten Eindruck.

Hersteller:
KINGSOFT
Fritz Schäfer
Schnackebusch 4
5106 Roetgen

# SID-Kurs

Um Low-, High- und Band-Pass-Filter geht es im vorletzten Teil unserer Serie zur Sound-Programmierung. Hier noch einmal die für diese Effekte zuständigen Adressen:

| | |
|---|---|
| SID-Basis | Adresse 54272 ($D400) |
| Filterfrequenz Low-Byte | Adresse 54293 ($D415) |
| Filterfrequenz High-Byte | Adresse 54294 ($D416) |
| Filter auf 1. Stimme: | Bit 0 in Adresse 54295 ($D417) |
| Filter auf 2. Stimme: | Bit 1 in Adresse 54295 ($D417) |
| Filter auf 3. Stimme: | Bit 2 in Adresse 54295 ($D417) |
| Filterresonanz: | Bits 4-7 in Adresse 54295 ($D417) |
| Lautstärke: | Bits 0-3 in Adresse 54296 ($D418) |
| Low-Pass-Filter: | Bit 4 in Adresse 54296 ($D418) |
| Band-Pass-Filter: | Bit 5 in Adresse 54296 ($D418) |
| High-Pass-Filter: | Bit 6 in Adresse 54296 ($D418) |

# ID - Werkstatt

Auch diesmal stellen wir Ihnen wieder zwei Programme vor. Aus dem Bereich der Unterhaltung das Programm MUSIKMASTER und für alle diejenigen, die sich des öfteren mit Reisekostenabrechnungen herumplagen müssen, das Programm SPESOQUICK. Da wir uns bei dieser Rubrik in erster Linie als Mittler verstehen, wollen wir auch gleich die Autoren selbst zu Wort kommen lassen.

Der Autor von Musikmaster beschreibt sein Programm wie folgt: "... Das Programm ermöglicht es, bis zu 1000 Noten mit Namen und Länge einzugeben und mit unterschiedlicher Geschwindigkeit in verschiedenen Klängen wiederzugeben. Diese Notensequenzen können dann auch auf Kassette oder Diskette gespeichert und geladen werden. Das Notenspektrum umfaßt 5 Oktaven. Wie die Noten einzugeben sind, geht aus der Anleitung des Programms hervor..."

Der Autor des Programms Spesoquick schreibt unter anderem: "... Die Idee zu diesem Programm, das ich Spesoquick genannt habe, kam mir bei Ihrem Programm Reisekostenabrechnung aus 8/85. Mir fiel in diesem Zusammenhang auf, daß es aber für den 64er noch kein Programm für eine Resekostenabrechnung gibt, wie sie ein Mitarbeiter nach einer Dienstreise machen muß; jedenfalls ist mir ein solches Programm nicht bekannt . Da ich selber als Versicherungskaufmann tätig bin, erstellte ich somit ein Programm für Reisekostenabrechnungen eines Innendienstmitarbeiters der deutschen Versicherungswirtschaft. Hierbei habe ich verschiedene Verkehrsmittel ebenso berücksichtigt, wie z.B. unterschiedliche Tagessätze nach gewissen Gehaltsabstufungen. Selbstverständlich kann das Programm leicht an die Gegebenheiten anderer Sparten angeglichen werden, da dann lediglich die Vergütungssätze ausgetauscht werden müßten, was ohne weiteres möglich ist...."

Uns bleibt an dieser Stelle nur noch, wieder darauf hinzuweisen, daß Werkstatt-Produkte von der Redaktion nicht betreut werden. Wir können daher auch keine Fragen zur Bedienung, zu Programmfehlern oder Ähnlichem beantworten. Wenn die Adresse im Programm veröffentlicht ist, können Sie direkt mit dem Autor Kontakt aufnehmen.

## HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise: Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:

Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter. Das Programm sollte in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden. Sie können allerdings auch eine andere Hochsprache (zum Beispiel PASCAL) verwenden, für die es möglich ist, selbständig lauffähige Compilate zu erzeugen. Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein. Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht. Senden Sie uns Ihre Programme bitte auf Kassette oder Diskette mit kommentiertem Listing und Beschreibung. Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlichtlich erschwert wurde, zukünftig ungeprüft zurücksenden. Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse: 1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0800-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF). 2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein. 3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein. 4. Die CTRL-Taste darf nicht benutzt werden. Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

# Hinweise zur Bedienung

1.) Entfernen Sie - bei ausgeschaltetem Rechner - eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD"INPUT 64",8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise ("Bitte Band zurückspulen" und so weiter . . .).

4.) Merken Sie sich fünf Kommandos:

- (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

- (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert. Von Diskette wird automatisch neu geladen.) - (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

- (CTRL) und (s) steht für SAVEn und funktioniert bei allen Program-men, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwendungsprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisun-gen - Sie haben eine Kopie des Programms.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

- (CTRL) und (b) ermöglicht einen Bildschirmausdruck - natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Das Programm wählt automatisch die richtige Geräteadresse (4, 5 oder 6) aus.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist.

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner "abstürzen". Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

---

## Bei Ladeproblemen:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft. Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben. Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

## Tonkopf selbst justieren

Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme/Wiedergabekopf justieren können. Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kasset-

tenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage. Legen Sie nun die zweite Seite von INPUT 64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung PRESS PLAY ON TAPE kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar – bei nicht total verstellter Spurlage – völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am besten markieren), und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenden Vermerk an den Verlag ein (Adresse siehe Impressum).

Besitzer der Ausgabe 6/85 können es sich einfacher machen. Das dort veröffentlichte Programm RECORDER-JUSTAGE macht die Einstellung des Daten-Recorders zum Kinderspiel.

## Listing Justage

```
700 rem  j u s t a g e  1.2
710 :
800 fori=49199to49410
810 read w
820 ps=ps+w
830 pokei,w
840 next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k"
970 sys49338
999 :
1000 rem  von 49199 bis 49410
1001 :
1010 data173, 13,220,169,217,174,4,220,172,5,220,141,14,220,48,44,56
1020 data102,88,36,89,48,12,144,10,165,88,133,90,169,128,133,88,133
1030 data91,192,121,144,4,224,115,176,7,169,0,133,92,56,176,11,165
1040 data92,73,128,133,92,36,92,16,19,24,102,88,36,89,48,12,144
1050 data10,165,88,133,90,169,128,133,88,133,91,104,168,104,170,104,64
1060 data96,36,91,16,252,132,91,165,90,96,160,128,132,89,165,88,201
1070 data22,208,250,132,88,160,10,132,89,132,91,36,91,16,252,132,91
1080 data165,90,201,22,208,226,136,208,241,32,133,192,201,22,240,249,96
1090 data32,147,252,120,32,23,248,165,1,41,31,133,1,133,192,169,47
1100 data141,20,3,169,192,141,21,3,169,127,141,13,220,169,144,141,13
1110 data220,173,17,208,41,239,141,17,208,169,70,141,4,220,169,129,141
1120 data5,220,88,32,142,192,201,42,208,249,173,32,208,41,15,168,200
1130 data140,32,208,76,237,192,208,76
ready.
```

# Am 20. Januar '86 an Ihrem Kiosk: INPUT 64 Ausgabe 1/86

## Wir bringen unter anderem:

### INPUT BASIC

Eine 16 KByte lange BASIC-Erweiterung mit den wichtigsten Hilfsprogrammen aus dem Magazin, wie: SuperTape, Hiresspeed (erweiterte Fassung), Sound-Befehlen, Sprite-Befehlen, PrintAt und Inkey, Renumber, Find, Delete, Auto und vielen neuen Befehlen: teilweises Abspeichern wählbarer Blöcke aus BASIC (nach Zeilennummern) oder von absoluten Adressbereichen, Print Using und so weiter. Mehr als 40 neue Befehle!

### Lohnsteuer-Berechnung

Dem Finanzamt auf die Finger schauen: Ihre nächste Lohnsteuererklärung machen Sie computerunterstützt. Orientiert an den Standard-Formularen und den aktuellen Steuertabellen. So wissen Sie schon am Jahresanfang, was Sie vom Staat zu erwarten haben.

### TextMagic

Mehr als ein Masken-Generator: ein Screen-Editor mit fast allen Funktionen einer Textverarbeitung, der für BASIC- und Assemblerprogrammierer das komfortable Erstellen von Bildschirmmasken erlaubt.

### Jahresinhalts-Datei

Das komplette und kommentierte Inhaltsverzeichnis der 85er INPUT 64-Ausgaben als Datei einer Mini-Dateiverarbeitung – die Sie natürlich auch für andere Zwecke einsetzen können.

### und außerdem:

Die lezte Folge des SID-Kurses und Soundcontrol dazu, Nico macht Physik zum Kinderspiel, Spiele und ein neues Rätsel.

---

## c't Magazin für Computertechnik

### Ausgabe 1/86 – jetzt am Kiosk

* Die Sprache C – Ende der Assembler-Programmierung? * 3D-Grafik auf dem C 64 * EPROM-Bank für den Atari ST * Applikation: Der neue Hitachi-Prozessor HD64180 * Das digitale Sound-Modul (DSM) im Live-Einsatz * Tests: TDI Pinaccle, ein schneller 68000-Rechner; Commodore Amiga * u.v.a.m

## elrad – Magazin für Elektronik

### Ausgabe 1/86 – ab 30.12. am Kiosk

* Die Hightech-Bauanleitung für erfahrene Hobbyelektroniker: Satelliten-TV-Empfangsanlage! Teil 1 in Heft 1/86 * Bauanleitungen: Combo-Verstärker, Noise-Gate, Batterie-Tester * Report: Überwachungstechnik '86 * u.v.a.m

## IMPRESSUM

**INPUT 64**
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 1
Tel.: (05 11) 53 52-0

---

**Technische Anfragen**
nur dienstags von 9-16.30 Uhr

---

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

**Herausgeber:** Christian Heise

**Redaktion:**
Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

**Ständige Mitarbeiter:**
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

**Vertrieb:** Anita Kreutzer

**Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:**

Verlag Heinz Heise GmbH
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

**Grafische Gestaltung:**
Wolfgang Ulber, Dirk Wollschläger

**Herstellung:** Heiner Niens

**Lithografie:**
Reprotechnik Hannover

**Druck:**
Leunisman GmbH, Hannover
CW Niemeyer Hameln

**Konfektionierung:**
Lettershop Brendler, Hannover

**Kassettenherstellung:**
SONOPRESS GMBH, Gütersloh

**INPUT 64** erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198,-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

**Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):**
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

**Verantwortlich:**
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61





**ISSN 0177-3771**

Titelidee **INPUT 64**
Titelfoto: digital design, Hannover

Titel – Grafik und – Musik:
Tim Pritlove
Farbian Rosenschein

hier abtrennen

# INPUT 64-Abonnement — Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat ____________

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.
○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.
(Bitte ankreuzen/Nichtzutreffendes streichen.)

## Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift ____________

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

# INPUT 64-Abonnement — Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl — Konto-Nr.

Geldinstitut — Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

**INPUT64**

**Vertriebsabteilung**
**Verlag Heinz Heise GmbH**
**Postfach 61 04 07**

**3000 Hannover 61**